BS ISO 16684-2:2014

BSI Standards Publication

Graphic technology — Extensible metadata platform (XMP)

Part 2: Description of XMP schemas using RELAX NG

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of ISO 16684-2:2014.

The UK participation in its preparation was entrusted to Technical Committee PAI/43, Graphic technology.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2014

ISBN 978 0 580 80114 3

ICS 35.240.30; 37.100.99

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 December 2014.

Amendments/corrigenda issued since publication

Date	Text affected

INTERNATIONAL STANDARD

ISO 16684-2

First edition
2014-12-01

Graphic technology — Extensible metadata platform (XMP) —

Part 2: Description of XMP schemas using RELAX NG

Technologie graphique — Plate-forme de métadonnées extensibles (XMP) —

Partie 2: Description des schémas XMP utilisant RELAX NG

Reference number
ISO 16684-2:2014(E)

COPYRIGHT PROTECTED DOCUMENT

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

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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 ISO documents 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 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 WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/130 *Graphic technology*.

ISO 16684 consists of the following parts, under the general title *Graphic technology — Extensible metadata platform (XMP)*:

— *Part 1: Data model, serialization, and core properties*

— *Part 2: Description of XMP schemas using RELAX NG*

Introduction

ISO 16684 (all parts) defines aspects of the Extensible Metadata Platform (XMP) that are generic, neutral to the domain of usage. Refer to the Introduction in ISO 16684-1 for general information. This part of ISO 16684 is about description of XMP schemas for formal or mechanical validation of XMP. RELAX NG has been chosen as the schema language. It is an ISO standard, ISO/IEC 19757-2, and is both powerful and easy to use.

There are two major components of formal validation, schemas and validation engines. A schema, or schema file, is a formal description of constraints regarding the structure and contents of properties in an XMP packet, on top of the requirements for conforming XMP packets as mandated by ISO 16684-1. A validation engine is a software tool that compares an input XMP packet to one or more schemas, and produces a report on whether the XMP packet conforms to the schemas.

This part of ISO 16684 defines policies for validation engines to follow so that schemas can be shared, so that the schemas do not require customization for each validation engine. It also defines policies for schemas to follow in order to operate with a conforming validation engine, and to make the schemas robust and modular.

This part of ISO 16684 does not address how a validation engine reports success or failure. Reporting success is easy, reporting failure can be complicated by a number of factors.

— It can be difficult to relate a specific RDF usage error to a human-understood XMP data model.

— Recovery from one error can be difficult, masking other errors after the first.

— As an open model, creation of new data items in XMP is expected. Allowing for this in schemas and/or clear reporting of unexpected input when validating can be difficult.

INTERNATIONAL STANDARD **ISO 16684-2:2014(E)**

Graphic technology — Extensible metadata platform (XMP) —

Part 2: Description of XMP schemas using RELAX NG

1 Scope

This part of ISO 16684 specifies the use of RELAX NG to describe serialized XMP metadata. This applies to how conforming schemas can use the features of RELAX NG.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 16684-1:2012, *Graphic technology — Extensible metadata platform (XMP) specification — Part 1: Data model, serialization and core properties*

ISO/IEC 19757-2, *Information technology — Document Schema Definition Language (DSDL) — Part 2: Regular-grammar-based validation — RELAX NG*

3 Terms and definitions

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

3.1
canonical serialization
serialization providing a one-to-one mapping between the XMP data model and the serialized XML

3.2
general qualifier
XMP qualifier other than xml:lang

3.3
schema
schema file
formal description of serialized XMP

3.4
validation
process of verifying whether serialized XMP follows one or more schemas

3.5
validation engine
software tool that performs validation

3.6
XMP
extensible metadata platform, as defined by ISO 16684-1

3.7
XMP entity
XMP property, array item, structure field, or qualifier

4 Conformance

This part of ISO 16684 describes methodology to create interoperable software and schema files in order to validate XMP metadata using regular-grammar-based validation schemas defined in ISO/IEC 19757-2 and referred to in this part of ISO 16684 as RELAX NG. Conformance on the part of a software validation engine enables the creation of schema files with lower complexity. Conforming validation engines shall adhere to all requirements of ISO 16684-1 and this part of ISO 16684. Conforming validation engines can provide additional features that are not explicitly forbidden by this part of ISO 16684. Conformance on the part of schema files enables their interchange among conforming validation engines. Conforming schema files shall adhere to all requirements of this part of ISO 16684. Conforming schema files can provide additional features that are not explicitly forbidden by this part of ISO 16684.

5 Canonical serialization of XMP

5.1 General

A major difficulty in validating XMP is that the RDF metadata format used by XMP allows multiple XML representations for the same metadata content. For reference, see ISO 16684-1:2012, 7.9. The RELAX NG schema language is used to validate serialized XML, not the RDF or XMP data models. Writing a RELAX NG schema to cover all possible XML forms for XMP is unacceptably complex. A canonical serialization of XMP is defined to limit this complexity. The canonical serialization requires specific forms of the XMP serialization defined in ISO 16684-1:2012, Clause 7, banning other forms. This provides a one-to-one mapping between the XMP data model and the canonical XML.

An XMP validation engine shall produce a canonical serialization of the XMP as part of the validation process. The validation engine shall accept any XMP as input that is allowed under ISO 16684-1. This XMP shall be parsed then serialized using the canonical forms described in this Clause. Next, the RELAX NG schema or schemas shall be applied to that canonical serialization. The XMP validation engine shall not modify original files containing the input XMP as part of the validation process.

5.2 XMP packet serialization

The canonical serialization of XMP shall have an rdf:RDF outermost XML element, which shall contain a single rdf:Description element to contain all XMP properties. That is, properties for all top-level namespaces are within this one rdf:Description element. The XMP properties shall be grouped by namespace. The namespace groups can have any order.

EXAMPLE An XMP metadata packet is shown containing XMP properties of different namespaces in a single rdf:Description element.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
              xmlns:dc="http://purl.org/dc/elements/1.1/"
              xmlns:xmp="http://ns.adobe.com/xap/1.0/"
              xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/">
       <rdf:Description rdf:about="">
              <dc:format>image/jpeg</dc:format>
              <xmp:Rating>3</xmp:Rating>
              <xmp:CreateDate>2012-02-29T12:33:44</xmp:CreateDate>
              <xmpMM:DocumentID>uuid:example1234</xmpMM:DocumentID>
       </rdf:Description>
</rdf:RDF>
```

5.3 Property serialization

The canonical serialization of XMP shall use the XML element form for XMP properties. The property attribute notation defined in ISO 16684-1:2012, 7.9.2.2 shall not be used.

5.4 Structure value serialization

The canonical serialization of XMP shall use a nested rdf:Description element for structure values. The structure fields shall be serialized as XML elements within that rdf:Description element. Values in a structure can be of type simple, structure, or array. They shall be serialized as described in 6.3, 6.4, and 6.5. The rdf:parseType=”resource” attribute notation defined in ISO 16684-1:2012, 7.9.2.3 shall not be used. The structure field attribute notation defined in ISO 16684-1:2012, 7.9.2.4 shall not be used.

EXAMPLE A structure value from an XMP metadata stream is shown in its canonically serialized form.

```
<xmpMM:DerivedFrom>
       <rdf:Description>
              <stRef:documentID>id:document</stRef:documentID>
              <stRef:instanceID>id:instance</stRef:instanceID>
       </rdf:Description>
</xmpMM:DerivedFrom>
```

5.5 Array value serialization

The canonical serialization of XMP shall use a nested rdf:Bag, rdf:Seq, or rdf:Alt element for array values. The array items shall be rdf:li elements within the rdf:Bag, rdf:Seq, or rdf:Alt element.

EXAMPLE An array value from an XMP metadata stream is shown in its canonically serialized form.

```
<dc:subject>
       <rdf:Bag>
              <rdf:li>subject 1</rdf:li>
       <rdf:li>subject 2</rdf:li>
       </rdf:Bag>
</dc:subject>
```

5.6 Qualifier serialization

The canonical serialization of XMP shall use the XML syntax defined in ISO 16684-1:2012, 7.8 for qualifiers. An xml:lang qualifier shall be serialized as an xml:lang attribute in the start tag of the XML element whose name is that of the XMP entity having the xml:lang qualifier. All general qualifiers shall be serialized as XML elements within an rdf:Description element that is within the XML element whose name is that of the XMP entity having the qualifier. The value of that XMP entity shall be within an rdf:value element within that rdf:Description element. The rdf:value element should be the first XML element within that rdf:Description element.

The rdf:parseType=”resource” notation defined in ISO 16684-1:2012, 7.9.2.3 shall not be used for an XMP entity with general qualifiers. The field attribute notation defined in ISO 16684-1:2012, 7.9.2.4 shall not be used for an XMP entity with general qualifiers. The RDF TypedNode notation defined in ISO 16684-1:2012, 7.9.2.5 shall not be used for an rdf:type qualifier.

EXAMPLE A part from an XMP metadata stream using a qualifier is shown in its canonically serialized form.

```
<xmp:Identifier>
       <rdf:Bag>
              <rdf:li>
                     <rdf:Description>
                            <rdf:value>0-13-110941-3</rdf:value>
                     <xmpidq:Scheme>ISBN</xmpidq:Scheme>
                     </rdf:Description>
       </rdf:li>
   </rdf:Bag>
</xmp:Identifier>
```

6 RELAX NG idioms for XMP

6.1 General

Having a canonical serialization for XMP is necessary to simplify the creation of RELAX NG schemas. As a consequence, considerably more freedom can be granted in the creation of the RELAX NG schemas.

This clause defines idioms for RELAX NG schemas that can improve reuse and interchange. They can also enable improved operation of a validation engine, for example making it easier to provide helpful error messages. Use of these idioms is not required.

6.2 Modularization

RELAX NG provides three main forms of modularization, the rng:define element, the rng:grammar element, and the rng:include element. The RELAX NG idioms for XMP provide recommendations for using these elements to simplify the creation of full RELAX NG schemas for XMP and to improve reuse and interchange of schema modules.

The rng:define element is perhaps the most important. It provides a means to create a named pattern, typically used to define a specific easily understood aspect of the XML to be validated. Lower level rng:define patterns can be referenced by name in higher level RELAX NG patterns.

The rng:grammar element provides a means to package rng:define patterns, and to control the scope of their names if desired.

The rng:include element provides a means to textually include one RELAX NG schema file within another.

While no specific naming conventions are required, the following approach can be chosen:

— names for individual properties can be composed as “prefix.localName”, e.g.”dc.title”;

— names for structure fields can be composed as “prefix.localName”, e.g. “stRef:documentID”;

— names for types can be composed as “Standard/Origin.Types.QValue.Typename”, e.g.”ISO 16684-1.Types.QValue.Boolean”, and “Standard/Origin.Types.Base.Typename”, e.g.”ISO 16684-1.Types.Base.Boolean”;

— names for array types can additionally denote the array type in their name; for example an unordered array of simple type Boolean can be expressed as e.g ISO 16684-1.Types.QValue.UnorderedArray.Boolean or ISO 16684-1.Types.Base.UnorderedArray.Boolean

NOTE The QValue idiom ensures that properties with or without qualifiers can be validated, the Base idiom contains the actual RELAX NG grammar for the type.

6.3 Use of data types

6.3.1 General

The XMP data model is defined in ISO 16684-1:2012, 6.3 using 3 forms of value: simple, structure, and array. Specific value types used are defined in ISO 16684-1:2012, 8.2. These can be thought of as data types, and formally defined that way for validation.

RELAX NG provides a primitive data type for unconstrained text. It also has features to describe various kinds of constrained text. It allows use of external data type libraries, most importantly the W3C XML Schema Datatypes. The rng:define element can be used to define arbitrary custom data types.

The rng:text type shall be used where an XMP value is unconstrained text. In all other cases, an rng:define element should be used to create a custom data type. This provides modularization of the data type semantics, makes the RELAX NG schema easier to understand, and fosters reuse of data types.

The pattern within an rng:define for a simple XMP value should describe the XML character data of the serialized value with appropriate semantic restrictions.

EXAMPLE Two rng:define instances describe constraints on text values to create custom data types.

```
<rng:define name=”ISO16684-1.Types.Boolean” combine=”choice”>
       <rng:data type=”string”>
              <rng:param name=”pattern”>True|False</rng:param>
       </rng:data>
</rng:define>
```



```
<rng:define name="xmp.type.Rating" combine="choice">
   <rng:choice>
      <rng:value type="float">-1</rng:value>
      <rng:data type="float">
         <rng:param name="minInclusive">0</rng:param>
         <rng:param name="maxInclusive">5</rng:param>
      </rng:data>
   </rng:choice>
</rng:define>
```

NOTE The combine="choice" attribute in the rng:define element is a workaround for the way rng:define and rng:include interact. Suppose the Boolean type is defined in one schema file, then included and used in two other schema files, which are then themselves included and used in another schema file. The final schema file will have two definitions for Boolean, which are, of course, identical. RELAX NG does not detect that they are identical and requires the combine attribute to be used telling how to select among them. Using combine="choice" says to pick any, which is appropriate since they are identical.

6.3.2 Structure value data types

The pattern within an rng:define for a structure value type shall describe the nested rdf:Description element and the structure fields within that. Each field should have a separate rng:define that describes its XML element and value. The value should be given as an rng:ref element referring to the appropriate data type.

EXAMPLE Several rng:define instances describe a structure value type using rng:ref elements.

```
<rng:define name="ISO16684-1.Types.ResourceRef" combine="choice">
       <rng:element name="rdf:Description">
              <rng:interleave>
                     <rng:optional>
                            <rng:ref name="stRef.documentID"/>
                     </rng:optional>
                     <rng:optional>
                            <rng:ref name="stRef.filePath"/>
                     </rng:optional>
              </rng:interleave>
       </rng:element>
</rng:define>

<rng:define name="stRef.documentID" combine="choice">
       <rng:element name="stRef:documentID">
              <rng:ref name="ISO16684-1.Types.GUID"/>
       </rng:element>
</rng:define>

<rng:define name="stRef.filePath" combine="choice">
       <rng:element name="stRef:filePath">
              <rng:ref name="ISO16684-1.Types.URI"/>
       </rng:element>
</rng:define>
```

NOTE Only two of the RenditionRef fields, defined in ISO 16684-1:2012, 8.2.2.9, are shown above in order to reduce the size of the example.

The rng:interleave element shall be used to denote that the structure fields can appear in any order. The rng:optional element shall be used if a structure field is optional, it shall be omitted if a structure field is required.

6.3.3 Array value data types

The pattern within an rng:define for an array value type shall describe the nested rdf:Bag, rdf:Seq, or rdf:Alt element, and the rdf:li array item elements within that.

EXAMPLE 1 An rng:define instance describes an array data type.

```
<rng:define name="ISO16684-1.Types.OrderedArray.Date" combine="choice">
       <rng:element name="rdf:Seq">
              <rng:zeroOrMore>
                     <rng:element name="rdf:li">
```

```
                        <rng:ref name="ISO16684-1.Types.Date"/>
                    </rng:element>
                </rng:zeroOrMore>
        </rng:element>
</rng:define>
```

The rng:zeroOrMore element should be used if the array can be empty and has no upper bound. The rng:oneOrMore element should be used if the array has at least one item and has no upper bound. For more than one required item, that number of explicit rdf:li patterns should be used. For a closed upper bound, the appropriate number of rdf:li patterns enclosed in rng:optional should be used.

EXAMPLE 2 An rng:define instance describes minimum and maximum number of entries in an array data type.

```
<rng:define name="ThreeToFiveOrderedDates" combine="choice">
        <rng:element name="rdf:Seq">
                <rng:ref name="DateItem"/>
                <rng:ref name="DateItem"/>
                <rng:ref name="DateItem"/>
                <rng:optional>
                        <rng:ref name="DateItem"/>
                </rng:optional>
                <rng:optional>
                        <rng:ref name="DateItem"/>
                </rng:optional>
        </rng:element>
</rng:define>

<rng:define name="DateItem" combine="choice">
        <rng:element name="rdf:li">
                <rng:ref name="ISO16684-1.Types.Date"/>
        </rng:element>
</rng:define>
```

6.3.4 Collections of data types

Collections of useful data types should be gathered in a RELAX NG schema file. This schema file should have an outer rng:grammar element that contains the rng:defines for the data types.

EXAMPLE An rng:grammar element in a separate schema file describes a collection of often used data types.

```
<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
                    datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

        <rng:define name="ISO16684-1.Types.Boolean" combine="choice">
                <rng:data type="string">
                        <rng:param name="pattern">True|False</rng:param>
                </rng:data>
        </rng:define>

        <rng:define name="ISO16684-1.Types.Integer" combine="choice">
                <rng:data type="integer"/>
        </rng:define>

        <rng:define name="ISO16684-1.Types.Rational" combine="choice">
                <rng:data type="string">
                        <rng:param name="pattern">\d+/[1-9]\d*</rng:param>
                </rng:data>
        </rng:define>

</rng:grammar>
```

NOTE Only three of the types defined in ISO 16684-1:2012, 8.2 are shown above in order to reduce the size of the example.

6.4 RELAX NG for properties

A top-level namespace shall have a RELAX NG schema file with an outer rng:grammar element. Within that shall be an rng:define element listing all of the properties it contains. Each property should have a

separate rng:define that describes its XML element and value. The value should be given as an rng:ref element referring to the appropriate data type.

EXAMPLE An outer rng:grammar element in its own file for a top-level namespace lists the rng:define elements for all of the properties it contains.

```
<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
                    xmlns:xmp="http://ns.adobe.com/xap/1.0/"
                    datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

   <rng:include href="ISO16684-1.Types.rng"/>

   <rng:define name="ISO16684-1.Properties-xmp" combine="choice">
       <rng:interleave>
              <rng:optional>
                     <rng:ref name="xmp.CreateDate"/>
              </rng:optional>
              <rng:optional>
                     <rng:ref name="xmp.Rating"/>
              </rng:optional>
       </rng:interleave>
   </rng:define>

   <rng:define name="xmp.CreateDate" combine="choice">
       <rng:element name="xmp:CreateDate">
              <rng:ref name="ISO16684-1.Types.Date"/>
       </rng:element>
   </rng:define>

   <rng:define name="xmp.Rating" combine="choice">
       <rng:element name="xmp:Rating">
              <rng:ref name="xmp.Rating"/>
       </rng:element>
   </rng:define>

   <rng:define name="xmp.type.Rating" combine="choice">
       <rng:choice>
              <rng:value type="float">-1</rng:value>
              <rng:data type="float">
                     <rng:param name="minInclusive">0</rng:param>
                     <rng:param name="maxInclusive">5</rng:param>
              </rng:data>
       </rng:choice>
   </rng:define>

</rng:grammar>
```

NOTE Only two of the XMP namespace properties defined in ISO 16684-1:2012, 8.4 are shown above in order to reduce the size of the example.

The rng:interleave element shall be used to denote that the properties can appear in any order. The rng:optional element shall be used if a property is optional, it shall be omitted if a property is required.

6.5 RELAX NG for XMP packets

A RELAX NG schema for a complete XMP packet shall have an outer rng:grammar element. This should contain rng:include elements for the allowed top-level namespaces and a single rng:start element. The rng:start element shall contain patterns for the outer rdf:RDF and rdf:Description elements of the canonical XMP serialization. Within the rdf:Description pattern shall be an rng:interleave element that contains references to the properties in the allowed namespaces.

EXAMPLE An outer rng:grammar element in its own file for a complete XMP packet contains rng:include elements for two top-level namespaces and an rng:start element with patterns for the outer rdf:RDF and rdf:Description elements.

```
<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

       <rng:include href="ISO16684-1_Properties-dc.rng"/>
```

```
   <rng:include href="ISO16684-1_Properties-xmp.rng"/>

   <rng:start>
       <rng:element name="rdf:RDF">
       <rng:element name="rdf:Description">
              <rng:attribute name="rdf:about"/>
            <rng:interleave>
             <rng:ref name="ISO16684-1.Properties-dc"/>
               <rng:ref name="ISO16684-1.Properties-xmp"/>
            </rng:interleave>
          </rng:element>
      </rng:element>
   </rng:start>

</rng:grammar>
```

6.6 RELAX NG for qualifiers

The use of an xml:lang qualifier in XMP metadata causes an xml:lang attribute to be added to the serialized XMP as defined in ISO 16684-1:2012, 7.8 and this Clause. To describe this in a RELAX NG schema, an rng:attribute element should be added to the data type rng:define pattern, where one exists. If there is no data type pattern, the rng:attribute element should be added to the XMP entity's rng:define pattern. An rng:optional element around the rng:attribute element shall be used if the qualifier is optional, it shall be omitted if the qualifier is required.

EXAMPLE 1 An rng:attribute in an rng:define pattern describes a language qualifier.

```
<rng:define name="LanguageAlternative" combine="choice">
       <rng:element name="rdf:Alt">
       <rng:zeroOrMore>
                     <rng:element name="rdf:li">
                            <rng:attribute name="xml:lang"><rng:text/></rng:attribute>
             <rng:text/>
                     </rng:element>
              </rng:zeroOrMore>
       </rng:element>
</rng:define>
```

The use of a general qualifier significantly changes the XML serialization of an XMP value, as defined in ISO 16684-1:2012, 7.8 and this Clause. This change is directly reflected in a RELAX NG schema, since the RELAX NG is describing the XML serialization. A data type rng:define should be used, which shall describe the nested rdf:Description element, the initial rdf:type element within that, and the general qualifier elements.

EXAMPLE 2 This example shows how general qualifiers impact the XML serialization of XMP values.

```
<rng:define name="xmp.IdentifierArray" combine="choice">
       <rng:element name="rdf:Bag">
              <rng:zeroOrMore>
                     <rng:element name="rdf:li">
                            <rng:ref name="xmp.IdentifierItem"/>
                     </rng:element>
              </rng:zeroOrMore>
       </rng:element>
</rng:define>

<rng:define name="xmp.IdentifierItem" combine="choice">
       <rng:element name="rdf:Description">
              <rng:element name="rdf:value" >
                     <rng:text/>
              </rng:element>
              <rng:ref name="xmpidq.Scheme"/>
       </rng:element>
</rng:define>

<rng:define name="xmpidq.Scheme" combine="choice">
       <rng:element name="xmpidq:Scheme">
              <rng:text/>
       </rng:element>
</rng:define>
```

In Example 2, the qualifier is required. If all qualifiers are optional, an rng:choice element shall be used to allow matching of either the unqualified or qualified forms.

EXAMPLE 3 This example shows the use of rng:choice elements to allow matching of either unqualified or qualified forms.

```
<rng:define name="xmp.IdentifierItem" combine="choice">
   <rng:choice>
      <rng:text/>   <!- The unqualified form. ->
      <rng:element name="rdf:Description"> <!- The qualified form. ->
         <rng:element name="rdf:value">
            <rng:text/>
         </rng:element>
         <rng:ref name="xmpidq.Scheme"/>
      </rng:element>
   </rng:choice>
</rng:define>
```

6.7 Extensions

All of the validation-related features of RELAX NG are in the schema as XML elements in the rng: namespace using the URI "http://relaxng.org/ns/structure/1.0". RELAX NG allows other XML elements and attributes to be in the schema, providing an open means of customization. This other XML does not impact the RELAX NG driven validation as long as it uses a namespace other than the RELAX NG namespace. This allows the schema to be used by other tools, or by a validation engine to provide additional features. One potential use of extension XML is to provide UI information.

6.8 Extension for providing UI information

This subclause defines XML for providing information that can be used to present an XMP schema, expressed in RELAX NG, in a user interface (UI). Such information can be provided for XMP schemas, for XMP properties and/or for XMP data types. It can be provided in one or more natural languages.

The namespace for providing user interface information for XMP schemas, XMP data types, and XMP properties is *xmlns:ui*="http://ns.iso.org/iso-16684-2/xmp-schema-ui-info/1.0". The entries shown in Table 1 are defined for this namespace:

Table 1 — Entries in the *ui* namespace

Entry	Usage
schema	Provides UI information for an XMP schema; can contain *label* and/or *description* entries.
property	Provides UI information for an XMP property; can contain *label* and/or *description* entries.
type	Provides UI information for an XMP data type; can contain *label* and/or *description* entries.
label	Contains text providing a human readable label for a schema, property, or type; should be qualified by a language attribute indicating the natural language in which the text is provided.
description	Contains text providing a human readable description or explanation for a schema, property, or type; should be qualified by a language attribute indicating the natural language in which the text is provided.

The following example illustrates how to include UI information for the Dublin Core property *dc:subject*:

EXAMPLE An rng:define element contains a label and descriptive information that can be used for presentation in a user interface.

```
<rng:define name="dc.subject" combine="choice">
   <rng:element name="dc:subject">
      <rng:ref name="ISO16684-1.Types.QValue.UnorderedArray.Text"/>
      <ui:property>
         <ui:label xml:lang="en" value=" List of keywords"/>
```

```
        <ui:description xml:lang="en"
             value=" List of descriptive phrases or keywords
                 that specify the content of the resource. Recommended best
                 practice is to use a controlled vocabulary."/>
        <ui:label xml:lang="de" value=" Liste von Schlagwörtern"/>
        <ui:description xml:lang="de"
             value="Liste beschreibender Sätze oder Schlagwörter, die den
                 Inhalt der Ressource beschreiben. Generell wird empfohlen,
                 ein kontrolliertes Vokabular zu verwenden."/>
      </ui:property>
   </rng:element>
</rng:define>
```

NOTE The RELAX NG schema for the Dublin Core namespace, provided as Annex B, makes use of the "Extension for providing UI information" as defined in this subclause, providing labels and descriptions in English and German.

7 RELAX NG schemas

Annexes A through F provide examples of RELAX NG schemas for various XMP data types, XMP namespaces, and XMP packets, with these examples being attached to this part of ISO 16684 as RELAX NG schema files.

— Annex A: RELAX NG schemas for Standard data types; attached as "ISO 16684-1_Types-Array.rng", "ISO 16684-1_Types-Simple.rng", and "ISO 16684-1_Types-Structure.rng"

— Annex B: RELAX NG schema for the Dublin Core namespace; attached as "ISO 16684-1_Properties-dc.rng"

— Annex C: RELAX NG schema for the XMP namespace; attached as "ISO 16684-1_Properties-xmp.rng"

— Annex D: RELAX NG schema for the XMP Media Management namespace; attached as "ISO 16684-1_Properties-xmpMM.rng"

— Annex E: RELAX NG schema for the XMP Rights Management namespace; attached as "ISO 16684-1_Properties-xmpRights.rng"

— Annex F: RELAX NG schema for a complete XMP packet; attached as "ISO 16684-1_CompletePacket.rng"

The RELAX NG schema files provided as Annexes A through F are text files and can be opened using a simple text editor.

The source code for these schemas can be freely used, in whole or in part, without any constraints. The source code can be used as the basis of derivative works. "Used" means that the source code is copied, modified, distributed in its original or a modified form, standalone or as part of something else, including open source code and projects, commercially or without charging fees.

Annex A
(informative)

RELAX NG schema for standard data types

A.1 Simple data types

```
<?xml version="1.0" encoding="utf-8"?>

<!--
       Core simple value types for XMP defined in ISO 16684-1 clause 8.2. All
       stored values are of course text, the types such as Integer or Boolean are
       the semantic interpretations.

       There are two patterns for each type, one named
       ISO16684-1.Types.QValue.Xyz and one named ISO16684-1.Types.Base.Xyz. The
       "Base" pattern defines the raw type. It should only be used in the "QValue"
       pattern, or in cases where use of XMP qualifiers is explicitly not allowed.
       The QValue pattern contains an rng:choice between the Base type and an
rdf:Description
       element with nested rdf:value element of the Base type. The second choice
       covers the use of unexpected XMP qualifiers. The qualifiers themselves would
       get validation errors, but not the qualified value.
-->

<rng:grammar
       xmlns:rng="http://relaxng.org/ns/structure/1.0"
       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:xmpidq="http://ns.adobe.com/xmp/Identifier/qual/1.0/"
       datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

       <!-- ================================================================= -->
       <!-- Standard qualifiers. Not types, but the most convenient location. -->
       <!-- ================================================================= -->

       <!-- xml:lang -->

       <rng:define name="ISO16684-1.Types.Qualifier.xml-lang"
              combine="choice">
              <rng:attribute name="xml:lang">
                     <rng:data type="string">
                            <rng:param name="pattern">x-default|([a-zA-Z]+(-[a-
zA-Z0-9]+)*)</rng:param>
                     </rng:data>
              </rng:attribute>
       </rng:define>

       <!-- xmpidq:Scheme -->

       <rng:define name="ISO16684-1.Types.Qualifier.xmpidq-Scheme"
              combine="choice">
              <rng:element name="xmpidq:Scheme">
                     <rng:ref name="ISO16684-1.Types.QValue.Text" />
              </rng:element>
       </rng:define>


       <!-- ================= -->
       <!-- General use types -->
       <!-- ================= -->

       <!-- Text -->

       <rng:define name="ISO16684-1.Types.QValue.Text" combine="choice">
```

```
            <rng:choice>
                  <rng:ref name="ISO16684-1.Types.Base.Text" />
                  <rng:element name="rdf:Description">
                        <rng:element name="rdf:value">
                              <rng:ref name="ISO16684-1.Types.Base.Text" />
                        </rng:element>
                  </rng:element>
            </rng:choice>
      </rng:define>

      <rng:define name="ISO16684-1.Types.Base.Text" combine="choice">
            <rng:text />
      </rng:define>

      <!-- Boolean -->

      <rng:define name="ISO16684-1.Types.QValue.Boolean" combine="choice">
            <rng:choice>
                  <rng:ref name="ISO16684-1.Types.Base.Boolean" />
                  <rng:element name="rdf:Description">
                        <rng:element name="rdf:value">
                              <rng:ref name="ISO16684-1.Types.Base.Boolean" />
                        </rng:element>
                  </rng:element>
            </rng:choice>
      </rng:define>

      <rng:define name="ISO16684-1.Types.Base.Boolean" combine="choice">
            <rng:data type="string">
                  <rng:param name="pattern">True|False</rng:param>
            </rng:data>
      </rng:define>

      <!-- Date -->

      <rng:define name="ISO16684-1.Types.QValue.Date" combine="choice">
            <rng:choice>
                  <rng:ref name="ISO16684-1.Types.Base.Date" />
                  <rng:element name="rdf:Description">
                        <rng:element name="rdf:value">
                              <rng:ref name="ISO16684-1.Types.Base.Date" />
                        </rng:element>
                  </rng:element>
            </rng:choice>
      </rng:define>

      <rng:define name="ISO16684-1.Types.Base.Date" combine="choice">
            <rng:data type="string">
                  <rng:param name="pattern">(\d{4}((-)?(0[1-9]|1[0-2])((-)?(0[1-9]|[1-2]
[0-9]|3[0-1])(T(24:00(:00(\.[0]+)?)?|(([0-1][0-9]|2[0-3])(:)[0-5][0-9])((:)[0-5][0-9]
(\.[\d]+)?)?)((\+|-)(14:00|(0[0-9]|1[0-3])(:)[0-5][0-9])|Z)?)?)?)?)</rng:param>
            </rng:data>
      </rng:define>

      <!-- Integer, real, and rational numbers -->

      <rng:define name="ISO16684-1.Types.QValue.Integer" combine="choice">
            <rng:choice>
                  <rng:ref name="ISO16684-1.Types.Base.Integer" />
                  <rng:element name="rdf:Description">
                        <rng:element name="rdf:value">
                              <rng:ref name="ISO16684-1.Types.Base.Integer" />
                        </rng:element>
                  </rng:element>
            </rng:choice>
      </rng:define>

      <rng:define name="ISO16684-1.Types.Base.Integer" combine="choice">
            <rng:data type="integer" />
      </rng:define>
```

```
<rng:define name="ISO16684-1.Types.QValue.Real" combine="choice">
	<rng:choice>
		<rng:ref name="ISO16684-1.Types.Base.Real" />
		<rng:element name="rdf:Description">
			<rng:element name="rdf:value">
				<rng:ref name="ISO16684-1.Types.Base.Real" />
			</rng:element>
		</rng:element>
	</rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.Real" combine="choice">
	<rng:data type="double" />
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.Rational" combine="choice">
	<rng:choice>
		<rng:ref name="ISO16684-1.Types.Base.Rational" />
		<rng:element name="rdf:Description">
			<rng:element name="rdf:value">
				<rng:ref name="ISO16684-1.Types.Base.Rational" />
			</rng:element>
		</rng:element>
	</rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.Rational" combine="choice">
	<rng:data type="string">
		<rng:param name="pattern">\d+/[1-9]\d*</rng:param>
	</rng:data>
</rng:define>


<!-- ====================== -->
<!-- More specialized types -->
<!-- ====================== -->

<!-- Agent name -->

<rng:define name="ISO16684-1.Types.QValue.AgentName"
	combine="choice">
	<rng:choice>
		<rng:ref name="ISO16684-1.Types.Base.AgentName" />
		<rng:element name="rdf:Description">
			<rng:element name="rdf:value">
				<rng:ref name="ISO16684-1.Types.Base.AgentName" />
			</rng:element>
		</rng:element>
	</rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.AgentName" combine="choice">
	<rng:text />
</rng:define>

<!-- GUID -->

<rng:define name="ISO16684-1.Types.QValue.GUID" combine="choice">
	<rng:choice>
		<rng:ref name="ISO16684-1.Types.Base.GUID" />
		<rng:element name="rdf:Description">
			<rng:element name="rdf:value">
				<rng:ref name="ISO16684-1.Types.Base.GUID" />
			</rng:element>
		</rng:element>
	</rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.GUID" combine="choice">
	<rng:text />
</rng:define>
```

```
	<!-- Locale -->

	<rng:define name="ISO16684-1.Types.QValue.Locale" combine="choice">
		<rng:choice>
			<rng:ref name="ISO16684-1.Types.Base.Locale" />
			<rng:element name="rdf:Description">
				<rng:element name="rdf:value">
					<rng:ref name="ISO16684-1.Types.Base.Locale" />
				</rng:element>
			</rng:element>
		</rng:choice>
	</rng:define>

	<rng:define name="ISO16684-1.Types.Base.Locale" combine="choice">
		<rng:data type="string">
			<rng:param name="pattern">[a-zA-Z]+(-[a-zA-Z0-9]+)*</rng:param>
		</rng:data>
	</rng:define>

	<!-- MIME-type -->

	<rng:define name="ISO16684-1.Types.QValue.MIMEType" combine="choice">
		<rng:choice>
			<rng:ref name="ISO16684-1.Types.Base.MIMEType" />
			<rng:element name="rdf:Description">
				<rng:element name="rdf:value">
					<rng:ref name="ISO16684-1.Types.Base.MIMEType" />
				</rng:element>
			</rng:element>
		</rng:choice>
	</rng:define>

	<rng:define name="ISO16684-1.Types.Base.MIMEType" combine="choice">
		<rng:data type="string">
			<rng:param
name="pattern">(application|audio|example|image|message|model|multipart|text|video)/
[a-zA-Z0-9]+([+\-.][a-zA-Z0-9]+)*</rng:param>
		</rng:data>
	</rng:define>

	<!-- Proper name -->

	<rng:define name="ISO16684-1.Types.QValue.ProperName"
		combine="choice">
		<rng:choice>
			<rng:ref name="ISO16684-1.Types.Base.ProperName" />
			<rng:element name="rdf:Description">
				<rng:element name="rdf:value">
					<rng:ref name="ISO16684-1.Types.Base.ProperName" />
				</rng:element>
			</rng:element>
		</rng:choice>
	</rng:define>

	<rng:define name="ISO16684-1.Types.Base.ProperName" combine="choice">
		<rng:text />
	</rng:define>

	<!-- Rendition class -->

	<rng:define name="ISO16684-1.Types.QValue.RenditionClass"
		combine="choice">
		<rng:choice>
			<rng:ref name="ISO16684-1.Types.Base.RenditionClass" />
			<rng:element name="rdf:Description">
				<rng:element name="rdf:value">
					<rng:ref name="ISO16684-1.Types.Base.RenditionClass" />
				</rng:element>
			</rng:element>
		</rng:choice>
```

```
        </rng:define>

        <rng:define name="ISO16684-1.Types.Base.RenditionClass"
               combine="choice">
               <rng:text />
        </rng:define>

        <!-- URI -->
        <rng:define name="ISO16684-1.Types.QValue.URI" combine="choice">
               <rng:choice>
                      <rng:ref name="ISO16684-1.Types.Base.URI" />
                      <rng:element name="rdf:Description">
                             <rng:element name="rdf:value">
                                    <rng:ref name="ISO16684-1.Types.Base.URI" />
                             </rng:element>
                      </rng:element>
               </rng:choice>
        </rng:define>

        <rng:define name="ISO16684-1.Types.Base.URI" combine="choice">
               <rng:text />
        </rng:define>

        <!-- URL -->

        <rng:define name="ISO16684-1.Types.QValue.URL" combine="choice">
               <rng:choice>
                      <rng:ref name="ISO16684-1.Types.Base.URL" />
                      <rng:element name="rdf:Description">
                             <rng:element name="rdf:value">
                                    <rng:ref name="ISO16684-1.Types.Base.URL" />
                             </rng:element>
                      </rng:element>
               </rng:choice>
        </rng:define>

        <rng:define name="ISO16684-1.Types.Base.URL" combine="choice">
               <rng:text />
        </rng:define>

</rng:grammar>
```

A.2 Array data types

```
<?xml version="1.0" encoding="utf-8"?>

<!--
Core array value types for XMP defined in ISO 16684-1 clause 8.2.

There are two patterns for each type, one named ISO16684-1.Types.QValue.Xyz and
one named ISO16684-1.Types.Base.Xyz. The "Base" pattern defines the raw type. It
should only be used in the "QValue" pattern, or in cases where use of XMP
qualifiers is explicitly not allowed. The QValue pattern contains an rng:choice
between the Base type and an rdf:Description element with nested rdf:value
element of the Base type. The second choice covers the use of unexpected XMP
qualifiers. The qualifiers themselves would get validation errors, but not the
qualified value.

All combinations are defined for arrays of the general simple types. Only the
necessary types are defined for more specialized arrays. The array item types
used here are all "QTypes", allowing for qualifiers.
-->

<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

   <!-- ================= -->
   <!-- General use types -->
   <!-- ================= -->
```

```
<!-- Unordered, ordered, alternative arrays of general text -->

<rng:define name="ISO16684-1.Types.QValue.UnorderedArray.Text" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Text"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Text"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.UnorderedArray.Text" combine="choice">
   <rng:element name="rdf:Bag">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Text"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.OrderedArray.Text" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Text"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Text"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.OrderedArray.Text" combine="choice">
   <rng:element name="rdf:Seq">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Text"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.AlternativeArray.Text" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Text"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Text"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.AlternativeArray.Text" combine="choice">
   <rng:element name="rdf:Alt">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Text"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<!-- Unordered, ordered, alternative arrays of Boolean -->

<rng:define name="ISO16684-1.Types.QValue.UnorderedArray.Boolean" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Boolean"/>
      <rng:element name="rdf:Description">
```

```
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Boolean"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.UnorderedArray.Boolean" combine="choice">
   <rng:element name="rdf:Bag">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Boolean"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.OrderedArray.Boolean" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Boolean"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Boolean"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.OrderedArray.Boolean" combine="choice">
   <rng:element name="rdf:Seq">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Boolean"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.AlternativeArray.Boolean" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Boolean"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Boolean"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.AlternativeArray.Boolean" combine="choice">
   <rng:element name="rdf:Alt">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Boolean"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<!-- Unordered, ordered, alternative arrays of Date -->

<rng:define name="ISO16684-1.Types.QValue.UnorderedArray.Date" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Date"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Date"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>
```

```
<rng:define name="ISO16684-1.Types.Base.UnorderedArray.Date" combine="choice">
   <rng:element name="rdf:Bag">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Date"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.OrderedArray.Date" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Date"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Date"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.OrderedArray.Date" combine="choice">
   <rng:element name="rdf:Seq">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Date"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.AlternativeArray.Date" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Date"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Date"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.AlternativeArray.Date" combine="choice">
   <rng:element name="rdf:Alt">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Date"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<!-- Unordered, ordered, alternative arrays of integer -->

<rng:define name="ISO16684-1.Types.QValue.UnorderedArray.Integer" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Integer"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Integer"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.UnorderedArray.Integer" combine="choice">
   <rng:element name="rdf:Bag">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Integer"/>
```

```
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.OrderedArray.Integer" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Integer"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Integer"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.OrderedArray.Integer" combine="choice">
   <rng:element name="rdf:Seq">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Integer"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.AlternativeArray.Integer" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Integer"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Integer"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.AlternativeArray.Integer" combine="choice">
   <rng:element name="rdf:Alt">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Integer"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<!-- Unordered, ordered, alternative arrays of real -->

<rng:define name="ISO16684-1.Types.QValue.UnorderedArray.Real" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Real"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Real"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.UnorderedArray.Real" combine="choice">
   <rng:element name="rdf:Bag">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Real"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.OrderedArray.Real" combine="choice">
```

```
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Real"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Real"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.OrderedArray.Real" combine="choice">
   <rng:element name="rdf:Seq">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Real"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.AlternativeArray.Real" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Real"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Real"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.AlternativeArray.Real" combine="choice">
   <rng:element name="rdf:Alt">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Real"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<!-- Unordered, ordered, alternative arrays of rational -->

<rng:define name="ISO16684-1.Types.QValue.UnorderedArray.Rational" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Rational"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Rational"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.UnorderedArray.Rational" combine="choice">
   <rng:element name="rdf:Bag">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Rational"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.OrderedArray.Rational" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Rational"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.OrderedArray.Rational"/>
         </rng:element>
```

```
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.OrderedArray.Rational" combine="choice">
   <rng:element name="rdf:Seq">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Rational"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="ISO16684-1.Types.QValue.AlternativeArray.Rational" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Rational"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.AlternativeArray.Rational"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.AlternativeArray.Rational" combine="choice">
   <rng:element name="rdf:Alt">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.QValue.Rational"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<!-- ====================== -->
<!-- More specialized types -->
<!-- ====================== -->

<!-- Language Alternative array -->

<rng:define name="ISO16684-1.Types.QValue.LanguageAlternative" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.LanguageAlternative"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.LanguageAlternative"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="ISO16684-1.Types.Base.LanguageAlternative" combine="choice">
   <rng:element name="rdf:Alt">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="ISO16684-1.Types.Qualifier.xml-lang"/>
            <rng:ref name="ISO16684-1.Types.QValue.Text"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<!-- Unordered array of Locale -->

<rng:define name="ISO16684-1.Types.QValue.UnorderedArray.Locale" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Locale"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.Locale"/>
```

```
            </rng:element>
         </rng:element>
      </rng:choice>
   </rng:define>

   <rng:define name="ISO16684-1.Types.Base.UnorderedArray.Locale" combine="choice">
      <rng:element name="rdf:Bag">
         <rng:zeroOrMore>
            <rng:element name="rdf:li">
               <rng:ref name="ISO16684-1.Types.QValue.Locale"/>
            </rng:element>
         </rng:zeroOrMore>
      </rng:element>
   </rng:define>

   <!-- Unordered and ordered arrays of ProperName -->

   <rng:define name="ISO16684-1.Types.QValue.UnorderedArray.ProperName" combine="choice">
      <rng:choice>
         <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.ProperName"/>
         <rng:element name="rdf:Description">
            <rng:element name="rdf:value">
               <rng:ref name="ISO16684-1.Types.Base.UnorderedArray.ProperName"/>
            </rng:element>
         </rng:element>
      </rng:choice>
   </rng:define>

   <rng:define name="ISO16684-1.Types.Base.UnorderedArray.ProperName" combine="choice">
      <rng:element name="rdf:Bag">
         <rng:zeroOrMore>
            <rng:element name="rdf:li">
               <rng:ref name="ISO16684-1.Types.QValue.ProperName"/>
            </rng:element>
         </rng:zeroOrMore>
      </rng:element>
   </rng:define>

   <rng:define name="ISO16684-1.Types.QValue.OrderedArray.ProperName" combine="choice">
      <rng:choice>
         <rng:ref name="ISO16684-1.Types.Base.OrderedArray.ProperName"/>
         <rng:element name="rdf:Description">
            <rng:element name="rdf:value">
               <rng:ref name="ISO16684-1.Types.Base.OrderedArray.ProperName"/>
            </rng:element>
         </rng:element>
      </rng:choice>
   </rng:define>

   <rng:define name="ISO16684-1.Types.Base.OrderedArray.ProperName" combine="choice">
      <rng:element name="rdf:Seq">
         <rng:zeroOrMore>
            <rng:element name="rdf:li">
               <rng:ref name="ISO16684-1.Types.QValue.ProperName"/>
            </rng:element>
         </rng:zeroOrMore>
      </rng:element>
   </rng:define>

</rng:grammar>
```

A.3 Structure data types

```
<?xml version="1.0" encoding="utf-8"?>

<!--
Core structure value types defined in ISO 16684-1 clause 8.2.

There are three patterns for each type, one named ISO16684-1.Types.QValue.Xyz,
one named ISO16684-1.Types.Base.Xyz, and one named ISO16684-1.Types.Fields.Xyz.
The "Base" pattern defines the raw type. It should only be used in the "QValue"
```

```
pattern, or in cases where use of XMP qualifiers is explicitly not allowed. The
QValue pattern contains an rng:choice between the Base type and an
rdf:Description element with nested rdf:value element of the Base type. The
second choice covers the use of unexpected XMP qualifiers. The qualifiers
themselves would get validation errors, but not the qualified value.

The Fields pattern is just a list of the fields. The Base pattern contains the
canonical rdf:Description element for the struct value, and a nested
rng:interleave element that in turn contains an rng:ref to the Fields pattern.
This allows derived struct types to contain more fields and still easily include
the Base fields.
-->

<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:stRef="http://ns.adobe.com/xap/1.0/sType/ResourceRef#"
             datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

   <!-- ResourceRef -->

   <rng:define name="ISO16684-1.Types.QValue.ResourceRef" combine="choice">
      <rng:choice>
         <rng:ref name="ISO16684-1.Types.Base.ResourceRef"/>
         <rng:element name="rdf:Description">
            <rng:element name="rdf:value">
               <rng:ref name="ISO16684-1.Types.Base.ResourceRef"/>
            </rng:element>
         </rng:element>
      </rng:choice>
   </rng:define>

   <rng:define name="ISO16684-1.Types.Base.ResourceRef" combine="choice">
      <rng:element name="rdf:Description">
         <rng:interleave>
            <rng:optional>
               <rng:ref name="stRef.documentID"/>
            </rng:optional>
            <rng:optional>
               <rng:ref name="stRef.filePath"/>
            </rng:optional>
            <rng:optional>
               <rng:ref name="stRef.instanceID"/>
            </rng:optional>
            <rng:optional>
               <rng:ref name="stRef.renditionClass"/>
            </rng:optional>
            <rng:optional>
               <rng:ref name="stRef.renditionParams"/>
            </rng:optional>
         </rng:interleave>
      </rng:element>
   </rng:define>

   <rng:define name="stRef.documentID" combine="choice">
      <rng:element name="stRef:documentID">
         <rng:ref name="ISO16684-1.Types.QValue.GUID"/>
      </rng:element>
   </rng:define>

   <rng:define name="stRef.filePath" combine="choice">
      <rng:element name="stRef:filePath">
         <rng:ref name="ISO16684-1.Types.QValue.URI"/>
      </rng:element>
   </rng:define>

   <rng:define name="stRef.instanceID" combine="choice">
      <rng:element name="stRef:instanceID">
         <rng:ref name="ISO16684-1.Types.QValue.GUID"/>
      </rng:element>
   </rng:define>
```

```
   <rng:define name="stRef.renditionClass" combine="choice">
      <rng:element name="stRef:renditionClass">
         <rng:ref name="ISO16684-1.Types.QValue.RenditionClass"/>
      </rng:element>
   </rng:define>

   <rng:define name="stRef.renditionParams" combine="choice">
      <rng:element name="stRef:renditionParams">
         <rng:ref name="ISO16684-1.Types.QValue.Text"/>
      </rng:element>
   </rng:define>

</rng:grammar>
```

Annex B
(informative)

RELAX NG schema for the Dublin Core namespace

```
<?xml version="1.0" encoding="utf-8"?>

<!—
       Property definitions for the Dublin Core namespace defined
       in ISO 16684-1 clause 8.3. There is an rng:define element
       for each top-level property providing the formal definition
       of that property. There is also an rng:define for the pattern
       "ISO16684-1.Properties-dc" that contains an rng:interleave
       of all of the top level properties. A full RelaxNG grammar
       would contain the outer rdf:RDF and rdf:Description elements,
       then refer to the relevant ISO16684-1.Properties-xx patterns
       for the necessary namespaces.
-->

<rng:grammar
              xmlns:rng="http://relaxng.org/ns/structure/1.0"
              xmlns:dc="http://purl.org/dc/elements/1.1/"
              xmlns:ui="http://ns.iso.org/iso-16684-2/xmp-schema-ui-info/1.0"
              datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

       <rng:include href="ISO16684-1_Types-Simple.rng" />
       <rng:include href="ISO16684-1_Types-Array.rng" />
       <rng:include href="ISO16684-1_Types-Structure.rng" />

       <!-- ======================================== -->
       <!-- Interleave of all Dublin Core properties. -->
       <!-- ======================================== -->

       <rng:define name="ISO16684-1.Properties-dc" combine="choice">
              <ui:schema>
                     <ui:label xml:lang="en" value="Dublin Core Metadata Element Set"/>
                     <ui:label xml:lang="de" value="Dublin Core Metadata Element Set"/>
                     <ui:description xml:lang="en" value="The Dublin Core Metadata Element
Set is a vocabulary of fifteen properties for use in resource description."/>
                     <ui:description xml:lang="de" value="Das Dublin Core Metadata Element
Set ist ein Wortschatz von fünfzehn Eigenschaften für die Beschreibung von Ressourcen."/>
              </ui:schema>
              <rng:interleave>
                     <rng:optional>
                            <rng:ref name="dc.contributor" />
                     </rng:optional>
                     <rng:optional>
                            <rng:ref name="dc.coverage" />
                     </rng:optional>
                     <rng:optional>
                            <rng:ref name="dc.creator" />
                     </rng:optional>
                     <rng:optional>
                            <rng:ref name="dc.date" />
                     </rng:optional>
                     <rng:optional>
                            <rng:ref name="dc.description" />
                     </rng:optional>
                     <rng:optional>
                            <rng:ref name="dc.format" />
                     </rng:optional>
                     <rng:optional>
                            <rng:ref name="dc.identifier" />
                     </rng:optional>
                     <rng:optional>
```

```
                        <rng:ref name="dc.language" />
                    </rng:optional>
                    <rng:optional>
                        <rng:ref name="dc.publisher" />
                    </rng:optional>
                    <rng:optional>
                        <rng:ref name="dc.relation" />
                    </rng:optional>
                    <rng:optional>
                        <rng:ref name="dc.rights" />
                    </rng:optional>
                    <rng:optional>
                        <rng:ref name="dc.source" />
                    </rng:optional>
                    <rng:optional>
                        <rng:ref name="dc.subject" />
                    </rng:optional>
                    <rng:optional>
                        <rng:ref name="dc.title" />
                    </rng:optional>
                    <rng:optional>
                        <rng:ref name="dc.type" />
                    </rng:optional>
                </rng:interleave>
        </rng:define>


        <!-- ============================================= -->
        <!-- Dublin Core properties in alpahbetical order. -->
        <!-- ============================================= -->

        <rng:define name="dc.contributor" combine="choice">
            <rng:element name="dc:contributor">
                <rng:ref name="ISO16684-1.Types.QValue.UnorderedArray.ProperName" />
                <ui:property>
                    <ui:label xml:lang="en" value="List of contributors"/>
                    <ui:description xml:lang="en" value="Entities responsible for
making contributions to the
                        resource. Examples of a contributor include a person, an
                        organization, or a service. Typically, the name of a
contributor should be
                        used to indicate the entity."/>
                    <ui:label xml:lang="de" value="Liste von Mitwirkenden"/>
                    <ui:description xml:lang="de" value="Entitäten, die dafür
verantwortlich sind, Beiträge zu der Ressource zu erschaffen. Ein Mitwirkende kann
beispielsweise eine Person, eine Organisation oder ein Dienst sein. Üblicherweise wird der
Name des Mitwirkenden zur Bezeichnung der Entität benutzt."/>
                </ui:property>
            </rng:element>
        </rng:define>

        <rng:define name="dc.coverage" combine="choice">
            <rng:element name="dc:coverage">
                <rng:ref name="ISO16684-1.Types.QValue.Text" />
                <ui:property>
                    <ui:label xml:lang="en" value="Coverage"/>
                    <ui:description xml:lang="en" value="The spatial or temporal
topic of the resource, the spatial applicability of the resource, or the jurisdiction
under which the resource is relevant. Spatial topic and spatial applicability may be a
named place or a location specified by its geographic coordinates. Temporal topic may be a
named period, date, or date range. A jurisdiction may be a named administrative entity or
a geographic place to which the resource applies. Recommended best practice is to use a
controlled vocabulary such as the Thesaurus of Geographic Names. Where appropriate, named
places or time periods can be used in preference to numeric identifiers such as sets of
coordinates or date ranges."/>
                    <ui:label xml:lang="de" value="Abdeckung"/>
                    <ui:description xml:lang="de" value="Der räumliche
oder zeitliche Bezug der Ressource, deren räumliche Anwendbarkeit oder der
Zuständigkeitsbereich, für den die Ressource gilt. Der räumliche Bezug und die räumliche
Anwendbarkeit können z. B. als ein benannter oder durch geographische Koordinaten
spezifizierter Ort ausgedrückt werden. Der zeitliche Bezug kann z. B. eine benannte
```

```
Periode, eine Zeitangabe oder ein Zeitraum sein. Ein Zuständigkeitsbereich kann eine
benannte Verwaltungseinheit oder ein geographischer Ort sein, für den die Ressource
zutrifft. Generell wird empfohlen, ein kontrolliertes Vokabular wie den Thesaurus of
Geographic Names zu verwenden. Des Weiteren können dort, wo es angemessen ist, benannte
Orte oder Zeitperioden gegenüber numerischen Identifikatoren, wie etwa Sätze von
Koordinaten oder Zeiträumen, bevorzugt werden.”/>
                        </ui:property>
                </rng:element>
        </rng:define>

        <rng:define name=”dc.creator” combine=”choice”>
                <rng:element name=”dc:creator”>
                        <rng:ref name=”ISO16684-1.Types.QValue.OrderedArray.ProperName” />
                        <ui:property>
                                <ui:label xml:lang=”en” value=”List of creators”/>
                                <ui:description xml:lang=”en” value=”An entity primarily
responsible for making the resource. Examples of a Creator include a person, an
organization, or a service. Typically, the name of a Creator should be used to indicate
the entity.”/>
                                <ui:label xml:lang=”de” value=”Liste der Ersteller”/>
                                <ui:description xml:lang=”de” value=”Liste der für die
Erstellung der Ressource hauptsächlich Verantwortlichen. Ein Ersteller kann beispielsweise
eine Person, eine Organisation oder ein Dienst sein. Üblicherweise wird hierbei der Name
des Erstellers verwendet.”/>
                        </ui:property>
                </rng:element>
        </rng:define>

        <rng:define name=”dc.date” combine=”choice”>
                <rng:element name=”dc:date”>
                        <rng:ref name=”ISO16684-1.Types.QValue.OrderedArray.Date” />
                        <ui:property>
                                <ui:label xml:lang=”en” value=”List of dates”/>
                                <ui:description xml:lang=”en” value=”Points or periods of time
associated with an event in the lifecycle of the resource. Date may be used to express
temporal information at any level of granularity. “/>
                                <ui:label xml:lang=”de” value=”Liste von Zeitangaben”/>
                                <ui:description xml:lang=”de” value=”Liste von Zeitpunkten oder
Zeitspannen, die in Verbindung mit einem Ereignis im Entwicklungsprozess der Ressource
stehen. Eine Zeitangabe kann verwendet werden, um zeitliche Informationen auf jeder
Granularitäts-Ebene auszudrücken.”/>
                        </ui:property>
                </rng:element>
        </rng:define>

        <rng:define name=”dc.description” combine=”choice”>
                <rng:element name=”dc:description”>
                        <rng:ref name=”ISO16684-1.Types.QValue.LanguageAlternative” />
                        <ui:property>
                                <ui:label xml:lang=”en” value=”Description”/>
                                <ui:description xml:lang=”en” value=”An account of the
resource. Description may include but is not limited to: an abstract, a table of contents,
a graphical representation, or a free-text account of the resource.”/>
                                <ui:label xml:lang=”de” value=”Beschreibung”/>
                                <ui:description xml:lang=”de” value=”Eine inhaltliche
Zusammenfassung der Ressource. Eine Beschreibung kann beispielsweise ein Abstract, ein
Inhaltsverzeichnis, eine graphische Darstellung oder eine Freitextbeschreibung der
Ressource sein, ist aber nicht darauf beschränkt.”/>
                        </ui:property>
                </rng:element>
        </rng:define>

        <rng:define name=”dc.format” combine=”choice”>
                <rng:element name=”dc:format”>
                        <rng:ref name=”ISO16684-1.Types.QValue.MIMEType” />
                        <ui:property>
                                <ui:label xml:lang=”en” value=”Format”/>
                                <ui:description xml:lang=”en” value=”The file format of the
resource, using a MIME type value.”/>
                                <ui:label xml:lang=”de” value=”Format”/>
```

```
                        <ui:description xml:lang="de" value="Das Dateiformat der
Ressource durch Angabe des MIME-Typs."/>
                    </ui:property>
                </rng:element>
        </rng:define>

        <rng:define name="dc.identifier" combine="choice">
                <rng:element name="dc:identifier">
                        <rng:ref name="ISO16684-1.Types.QValue.Text" />
                        <ui:property>
                                <ui:label xml:lang="en" value="Identifier"/>
                                <ui:description xml:lang="en" value="An unambiguous reference
to the resource within a given context. Recommended best practice is to identify the
resource by means of a string conforming to a formal identification system."/>
                                <ui:label xml:lang="de" value="Identifikator"/>
                                <ui:description xml:lang="de" value="Ein eindeutiger Hinweis
auf die Ressource in einem gegebenen Kontext. Generell wird empfohlen, die Ressource
mittels einer Zeichenkette, die einem formell anerkannten Identifikationssystem entspricht,
zu identifizieren."/>
                        </ui:property>
                </rng:element>
        </rng:define>

        <rng:define name="dc.language" combine="choice">
                <rng:element name="dc:language">
                        <rng:ref name="ISO16684-1.Types.QValue.UnorderedArray.Locale" />
                        <ui:property>
                                <ui:label xml:lang="en" value="List of languages"/>
                                <ui:description xml:lang="en" value="List of languages used in
the content of the resource. Recommended best practice is to use a controlled vocabulary
such as RFC 4646."/>
                                <ui:label xml:lang="de" value="Liste der Sprachen"/>
                                <ui:description xml:lang="de" value="Liste der im Inhalt der
Ressource verwendeten Sprachen. Generell wird empfohlen, ein kontrolliertes Vokabular wie
z. B. RFC 4646 zu benutzen."/>
                        </ui:property>
                </rng:element>
        </rng:define>

        <rng:define name="dc.publisher" combine="choice">
                <rng:element name="dc:publisher">
                        <rng:ref name="ISO16684-1.Types.QValue.UnorderedArray.ProperName" />
                        <ui:property>
                                <ui:label xml:lang="en" value="List of publishers"/>
                                <ui:description xml:lang="en" value="A Publisher is an entity
responsible for making the resource available. Examples of a Publisher include a person,
an organization, or a service. Typically, the name of a Publisher should be used to
indicate the entity."/>
                                <ui:label xml:lang="de" value="Liste der Verleger"/>
                                <ui:description xml:lang="de" value="Ein Verleger ist
eine Entität, die dafür verantworlich ist, eine Ressource verfügbar zu machen. Ein/e
Verleger/in kann beispielsweise eine Person, eine Organisation oder ein Dienst sein.
Üblicherweise wird der Name des Verlegers zur Bezeichnung der Entität benutzt."/>
                        </ui:property>
                </rng:element>
        </rng:define>

        <rng:define name="dc.relation" combine="choice">
                <rng:element name="dc:relation">
                        <rng:ref name="ISO16684-1.Types.QValue.UnorderedArray.Text" />
                </rng:element>
        </rng:define>

        <rng:define name="dc.rights" combine="choice">
                <rng:element name="dc:rights">
                        <rng:ref name="ISO16684-1.Types.QValue.LanguageAlternative" />
                        <ui:property>
                                <ui:label xml:lang="en" value="Rights"/>
                                <ui:description xml:lang="en" value="Information about rights held
in and over the resource. Typically, rights information includes a statement about various
property rights associated with the resource, including intellectual property rights."/>
```

```
                        <ui:label xml:lang=”de” value=”Rechteverwaltung”/>
                        <ui:description xml:lang=”de” value=”Informationen über
Rechte an der Ressource. Üblicherweise beinhalten Rechteinformationen Aussagen über
verschiedenartige Eigentumsrechte, die mit der Ressource verbunden sind, einschließlich
der Schutz- und Urheberrechte.”/>
                </ui:property>
        </rng:element>
    </rng:define>

    <rng:define name=”dc.source” combine=”choice”>
        <rng:element name=”dc:source”>
            <rng:ref name=”ISO16684-1.Types.QValue.Text” />
            <ui:property>
                <ui:label xml:lang=”en” value=”Source”/>
                <ui:description xml:lang=”en” value=”A related resource from
which the described resource is derived. The described resource may be derived from the
related resource in whole or in part. Recommended best practice is to identify the related
resource by means of a string conforming to a formal identification system.”/>
                <ui:label xml:lang=”de” value=”Quelle”/>
                <ui:description xml:lang=”de” value=”Eine Ressource, von
der die beschriebene Ressource abgeleitet ist. Die beschriebene Ressource kann von der
ursprünglichen Quelle vollständig oder teilweise abgeleitet sein. Generell wird empfohlen,
die ursprüngliche Quelle mittels einer Zeichenkette, die einem formell anerkannten
Identifikationssystem entspricht, zu identifizieren.”/>
            </ui:property>
        </rng:element>
    </rng:define>

    <rng:define name=”dc.subject” combine=”choice”>
        <rng:element name=”dc:subject”>
            <rng:ref name=”ISO16684-1.Types.QValue.UnorderedArray.Text” />
            <ui:property>
                <ui:label xml:lang=”en” value=”List of keywords”/>
                <ui:description xml:lang=”en” value=”List of descriptive
phrases or keywords that specify the content of the resource. Recommended best practice is
to use a controlled vocabulary. “/>
                <ui:label xml:lang=”de” value=”Liste von Schlagwörtern”/>
                <ui:description xml:lang=”de” value=”Liste beschreibender Sätze
oder Schlagwörter, die den Inhalt der Ressource beschreiben. Generell wird empfohlen, ein
kontrolliertes Vokabular zu verwenden.”/>
            </ui:property>
        </rng:element>
    </rng:define>

    <rng:define name=”dc.title” combine=”choice”>
        <rng:element name=”dc:title”>
            <rng:ref name=”ISO16684-1.Types.QValue.LanguageAlternative” />
            <ui:property>
                <ui:label xml:lang=”en” value=”Title”/>
                <ui:description xml:lang=”en” value=”A name given to the
resource.”/>
                <ui:label xml:lang=”de” value=”Titel”/>
                <ui:description xml:lang=”de” value=”Ein der Ressource
gegebener Name.”/>
            </ui:property></rng:element>
    </rng:define>

    <rng:define name=”dc.type” combine=”choice”>
        <rng:element name=”dc:type”>
            <rng:ref name=”ISO16684-1.Types.QValue.UnorderedArray.Text” />
            <ui:property>
                <ui:label xml:lang=”en” value=”Type”/>
                <ui:description xml:lang=”en” value=”The nature or genre of the
resource. Recommended best practice is to use a controlled vocabulary such as the DCMI
Type Vocabulary.”/>
                <ui:label xml:lang=”de” value=”Typ”/>
                <ui:description xml:lang=”de” value=”Art oder Genre der
Ressource. Generell wird empfohlen, ein kontrolliertes Vokabular wie z.B. das ‘DCMI Type
Vocabulary’ zu benutzen.”/>
            </ui:property>
        </rng:element>
```

```
        </rng:define>
</rng:grammar>
```

Annex C
(informative)

RELAX NG schema for the XMP namespace

```
<?xml version="1.0" encoding="utf-8"?>

<!--
Property definitions for the XMP namespace defined in ISO 16684-1 clause 8.4.
There is an rng:define element for each top-level property providing the formal
definition of that property. There is also an rng:define for the pattern
"ISO16684-1.Properties-xmp" that contains an rng:interleave of all of the top
level properties. A full RelaxNG grammar would contain the outer rdf:RDF and
rdf:Description elements, then refer to the relevant ISO16684-1.Properties-xx
patterns for the necessary namespaces.
-->

<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:xmp="http://ns.adobe.com/xap/1.0/"
             datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

   <rng:include href="ISO16684-1_Types-Simple.rng"/>
   <rng:include href="ISO16684-1_Types-Array.rng"/>
   <rng:include href="ISO16684-1_Types-Structure.rng"/>

   <!-- ================================= -->
   <!-- Interleave of all XMP properties. -->
   <!-- ================================= -->

   <rng:define name="ISO16684-1.Properties-xmp" combine="choice">
      <rng:interleave>
         <rng:optional>
            <rng:ref name="xmp.CreateDate"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmp.CreatorTool"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmp.Identifier"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmp.Label"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmp.MetadataDate"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmp.ModifyDate"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmp.Rating"/>
         </rng:optional>
      </rng:interleave>
   </rng:define>

   <!-- ================================== -->
   <!-- XMP properties in alpahbetical order. -->
   <!-- ================================== -->

   <rng:define name="xmp.CreateDate" combine="choice">
      <rng:element name="xmp:CreateDate">
         <rng:ref name="ISO16684-1.Types.QValue.Date"/>
      </rng:element>
   </rng:define>
```

```
<rng:define name="xmp.CreatorTool" combine="choice">
   <rng:element name="xmp:CreatorTool">
      <rng:ref name="ISO16684-1.Types.QValue.AgentName"/>
   </rng:element>
</rng:define>

<rng:define name="xmp.Identifier" combine="choice">
   <rng:element name="xmp:Identifier">
      <rng:ref name="xmp.QValue.IdentifierArray"/>
   </rng:element>
</rng:define>

<rng:define name="xmp.Label" combine="choice">
   <rng:element name="xmp:Label">
      <rng:ref name="ISO16684-1.Types.QValue.Text"/>
   </rng:element>
</rng:define>

<rng:define name="xmp.MetadataDate" combine="choice">
   <rng:element name="xmp:MetadataDate">
      <rng:ref name="ISO16684-1.Types.QValue.Date"/>
   </rng:element>
</rng:define>

<rng:define name="xmp.ModifyDate" combine="choice">
   <rng:element name="xmp:ModifyDate">
      <rng:ref name="ISO16684-1.Types.QValue.Date"/>
   </rng:element>
</rng:define>

<rng:define name="xmp.Rating" combine="choice">
   <rng:element name="xmp:Rating">
      <rng:ref name="xmp.QValue.Rating"/>
   </rng:element>
</rng:define>

<!-- ================================== -->
<!-- Local types for the XMP namespace. -->
<!-- ================================== -->

<rng:define name="xmp.QValue.IdentifierArray" combine="choice">
   <!-- Unordered array of text, items have optional xmpidq:Scheme qualifier. -->
   <rng:choice>
      <rng:ref name="xmp.Base.IdentifierArray"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="xmp.Base.IdentifierArray"/>
         </rng:element>
      </rng:element>
   </rng:choice>
</rng:define>

<rng:define name="xmp.Base.IdentifierArray" combine="choice">
   <rng:element name="rdf:Bag">
      <rng:zeroOrMore>
         <rng:element name="rdf:li">
            <rng:ref name="xmp.QValue.IdentifierItem"/>
         </rng:element>
      </rng:zeroOrMore>
   </rng:element>
</rng:define>

<rng:define name="xmp.QValue.IdentifierItem" combine="choice">
   <rng:choice>
      <rng:ref name="ISO16684-1.Types.Base.Text"/>
      <rng:element name="rdf:Description">
         <rng:element name="rdf:value">
            <rng:ref name="ISO16684-1.Types.Base.Text"/>
         </rng:element>
         <rng:ref name="ISO16684-1.Types.Qualifier.xmpidq-Scheme"/>
```

```
         </rng:element>
      </rng:choice>
   </rng:define>

   <rng:define name="xmp.QValue.Rating" combine="choice">
      <rng:choice>
         <rng:ref name="xmp.Base.Rating"/>
         <rng:element name="rdf:Description">
            <rng:element name="rdf:value">
               <rng:ref name="xmp.Base.Rating"/>
            </rng:element>
         </rng:element>
      </rng:choice>
   </rng:define>

   <rng:define name="xmp.Base.Rating" combine="choice">
      <rng:choice>
         <rng:value type="float">-1</rng:value>
         <rng:data type="float">
            <rng:param name="minInclusive">0</rng:param>
            <rng:param name="maxInclusive">5</rng:param>
         </rng:data>
      </rng:choice>
   </rng:define>

</rng:grammar>
```

Annex D
(informative)

RELAX NG schema for the XMP Media Management namespace

```
<?xml version="1.0" encoding="utf-8"?>

<!--
Property definitions for the XMP Media Management namespace defined in ISO
16684-1 clause 8.6. There is an rng:define element for each top-level property
providing the formal definition of that property. There is also an rng:define
for the pattern "ISO16684-1.Properties-xmpMM" that contains an rng:interleave of
all of the top level properties. A full RelaxNG grammar would contain the outer
rdf:RDF and rdf:Description elements, then refer to the relevant
ISO16684-1.Properties-xx patterns for the necessary namespaces.
-->

<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
             xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/"
             datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

   <rng:include href="ISO16684-1_Types-Simple.rng"/>
   <rng:include href="ISO16684-1_Types-Array.rng"/>
   <rng:include href="ISO16684-1_Types-Structure.rng"/>

   <!-- ================================================== -->
   <!-- Interleave of all XMP Media Management properties. -->
   <!-- ================================================== -->

   <rng:define name="ISO16684-1.Properties-xmpMM" combine="choice">
      <rng:interleave>
         <rng:optional>
            <rng:ref name="xmpMM.DerivedFrom"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpMM.DocumentID"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpMM.InstanceID"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpMM.OriginalDocumentID"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpMM.RenditionClass"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpMM.RenditionParams"/>
         </rng:optional>
      </rng:interleave>
   </rng:define>

   <!-- ================================================== -->
   <!-- XMP Media Management properties in alpahbetical order. -->
   <!-- ================================================== -->

   <rng:define name="xmpMM.DerivedFrom" combine="choice">
      <rng:element name="xmpMM:DerivedFrom">
         <rng:ref name="ISO16684-1.Types.QValue.ResourceRef"/>
      </rng:element>
   </rng:define>

   <rng:define name="xmpMM.DocumentID" combine="choice">
      <rng:element name="xmpMM:DocumentID">
         <rng:ref name="ISO16684-1.Types.QValue.GUID"/>
```

```
      </rng:element>
   </rng:define>

   <rng:define name="xmpMM.InstanceID" combine="choice">
      <rng:element name="xmpMM:InstanceID">
         <rng:ref name="ISO16684-1.Types.QValue.GUID"/>
      </rng:element>
   </rng:define>

   <rng:define name="xmpMM.OriginalDocumentID" combine="choice">
      <rng:element name="xmpMM:OriginalDocumentID">
         <rng:ref name="ISO16684-1.Types.QValue.GUID"/>
      </rng:element>
   </rng:define>

   <rng:define name="xmpMM.RenditionClass" combine="choice">
      <rng:element name="xmpMM:RenditionClass">
         <rng:ref name="ISO16684-1.Types.QValue.RenditionClass"/>
      </rng:element>
   </rng:define>

   <rng:define name="xmpMM.RenditionParams" combine="choice">
      <rng:element name="xmpMM:RenditionParams">
         <rng:ref name="ISO16684-1.Types.QValue.Text"/>
      </rng:element>
   </rng:define>

</rng:grammar>
```

Annex E
(informative)

RELAX NG schema for the XMP Rights Management namespace

```
<?xml version="1.0" encoding="utf-8"?>

<!--
Property definitions for the XMP Rights Management namespace defined in ISO
16684-1 clause 8.5. There is an rng:define element for each top-level property
providing the formal definition of that property. There is also an rng:define
for the pattern "ISO16684-1.Properties-xmpRights" that contains an
rng:interleave of all of the top level properties. A full RelaxNG grammar would
contain the outer rdf:RDF and rdf:Description elements, then refer to the
relevant ISO16684-1.Properties-xx patterns for the necessary namespaces.
-->

<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
             xmlns:xmpRights="http://ns.adobe.com/xap/1.0/rights/"
             datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

   <rng:include href="ISO16684-1_Types-Simple.rng"/>
   <rng:include href="ISO16684-1_Types-Array.rng"/>
   <rng:include href="ISO16684-1_Types-Structure.rng"/>

   <!-- ================================================== -->
   <!-- Interleave of all XMP Rights Management properties. -->
   <!-- ================================================== -->

   <rng:define name="ISO16684-1.Properties-xmpRights" combine="choice">
      <rng:interleave>
         <rng:optional>
            <rng:ref name="xmpRights.Certificate"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpRights.Marked"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpRights.Owner"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpRights.UsageTerms"/>
         </rng:optional>
         <rng:optional>
            <rng:ref name="xmpRights.WebStatement"/>
         </rng:optional>
      </rng:interleave>
   </rng:define>

   <!-- ======================================================== -->
   <!-- XMP Rights Management properties in alpahbetical order. -->
   <!-- ======================================================== -->

   <rng:define name="xmpRights.Certificate" combine="choice">
      <rng:element name="xmpRights:Certificate">
         <rng:ref name="ISO16684-1.Types.QValue.Text"/>
      </rng:element>
   </rng:define>

   <rng:define name="xmpRights.Marked" combine="choice">
      <rng:element name="xmpRights:Marked">
         <rng:ref name="ISO16684-1.Types.QValue.Boolean"/>
      </rng:element>
   </rng:define>
```

```
   <rng:define name="xmpRights.Owner" combine="choice">
      <rng:element name="xmpRights:Owner">
         <rng:ref name="ISO16684-1.Types.QValue.UnorderedArray.ProperName"/>
      </rng:element>
   </rng:define>

   <rng:define name="xmpRights.UsageTerms" combine="choice">
      <rng:element name="xmpRights:UsageTerms">
         <rng:ref name="ISO16684-1.Types.QValue.LanguageAlternative"/>
      </rng:element>
   </rng:define>

   <rng:define name="xmpRights.WebStatement" combine="choice">
      <rng:element name="xmpRights:WebStatement">
         <rng:ref name="ISO16684-1.Types.QValue.Text"/>
      </rng:element>
   </rng:define>

</rng:grammar>
```

Annex F
(informative)

RELAX NG schema for a complete XMP packet

```
<?xml version="1.0" encoding="utf-8"?>

<rng:grammar xmlns:rng="http://relaxng.org/ns/structure/1.0"
             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

   <rng:include href="ISO16684-1_Properties-dc.rng"/>
   <rng:include href="ISO16684-1_Properties-xmp.rng"/>
   <rng:include href="ISO16684-1_Properties-xmpMM.rng"/>
   <rng:include href="ISO16684-1_Properties-xmpRights.rng"/>

   <rng:start>

      <rng:element name="rdf:RDF">

         <rng:element name="rdf:Description">
            <rng:attribute name="rdf:about"/>

            <rng:interleave>
               <rng:ref name="ISO16684-1.Properties-dc"/>
               <rng:ref name="ISO16684-1.Properties-xmp"/>
               <rng:ref name="ISO16684-1.Properties-xmpMM"/>
               <rng:ref name="ISO16684-1.Properties-xmpRights"/>
            </rng:interleave>

         </rng:element>

      </rng:element>

   </rng:start>

</rng:grammar>
```

Bibliography

[1] *RDF/XML Syntax Specification* (Revised), W3C Recommendation 10 February 2004 http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210

[2] *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation 26 November 2008 http://www.w3.org/TR/2008/REC-xml-20081126

[3] *Namespaces in XML 1.0 (Second Edition)*, W3C Recommendation 16 August 2006 http://www.w3c.org/TR/2006/REC-xml-names-20060816

[4] *XML Schema Part 2: Datatypes (Second Edition)*, W3C Recommendation 28 October 2004 http://www.w3c.org/TR/2004/REC-xmlschema-2-20041028

ICS 35.240.30; 37.100.99
Price based on 39 pages

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

bsi.

...making excellence a habit.™