INTERNATIONAL STANDARD

ISO/IEC 21000-7

First edition
2004-10-15

AMENDMENT 1
2006-04-01

Information technology — Multimedia framework (MPEG-21) —

Part 7: Digital Item Adaptation

AMENDMENT 1: DIA Conversions and Permissions

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

Partie 7: Adaptation d'article numérique

AMENDEMENT 1: Conversions et droits d'accès DIA

Reference number
ISO/IEC 21000-7:2004/Amd.1:2006(E)

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

Published in Switzerland

Foreword

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

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

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

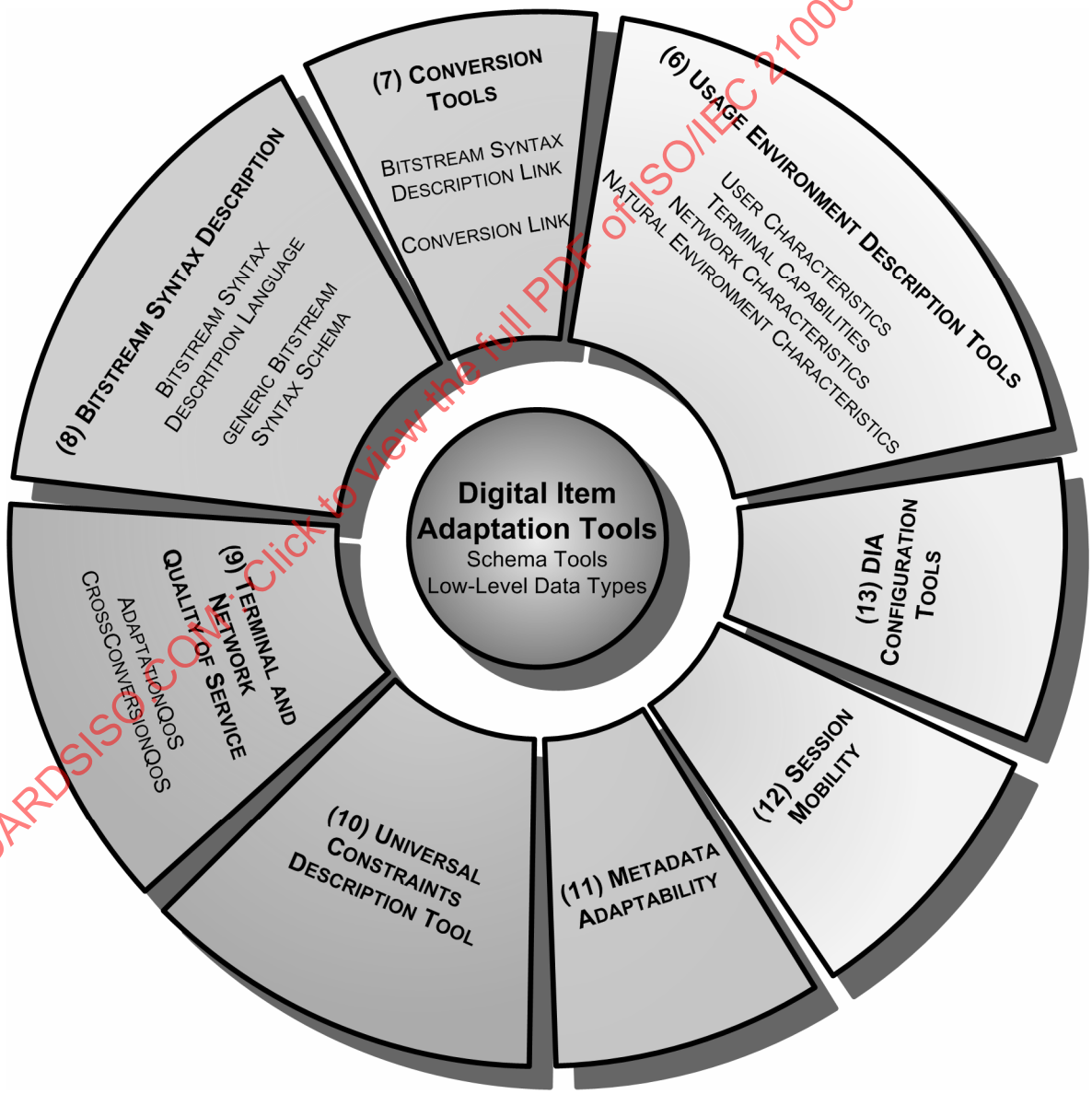
Amendment 1 to ISO/IEC 21000-7:2004 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Multimedia framework (MPEG-21) —

Part 7:
Digital Item Adaptation

AMENDMENT 1: DIA Conversions and Permissions

In subclause 1.4, replace Figure 2 with the following figure:



In subclause 1.4, replace the following text:

The second category is referred to as *BSDLink* which provides the facilities to create a rich variety of adaptation architectures based on tools specified within this part of ISO/IEC 21000, ISO/IEC 21000-2, and ISO/IEC 15398 among others. The syntax and semantics of this tool is specified in clause 7.

with:

The second category is referred to as *Conversion Tools* which provides the facilities to create a rich variety of adaptation architectures based on tools specified within this part of ISO/IEC 21000, ISO/IEC 21000-2, and ISO/IEC 15398 among others. The syntax and semantics of this tool is specified in clause 7.

In subclause 1.5, remove the following text:

This specification deals with adaptation but specifically does not address the relationship of rights and permissions to adaptations. The relationship of rights and permissions is to be addressed in an amendment to this part of ISO/IEC 21000. It is expected that users of this part of ISO/IEC 21000 will register terms describing their specific adaptations with the Registration Authority described in ISO/IEC 21000-6 in order to provide interoperability.

Append a new paragraph at the end of this subclause with the following text:

ISO/IEC 21000-5 and ISO/IEC 21000-6 provide the tools to permit playing, modifying, and adapting by controlling the kinds of things that can be changed. This part of ISO/IEC 21000 provides tools for use with ISO/IEC 21000-5 to provide the means by which that control over the changes that can occur when playing, modifying, or adapting digital items and their component resources can be effected. It is expected that users of this part of ISO/IEC 21000 will register terms describing their specific adaptations with the Registration Authority described in ISO/IEC 21000-6 in order to provide interoperability.

In subclause 3.4, in Table 1, append the following three additional rows:

r	urn:mpeg:mpeg21:2003:01-REL-R-NS
sx	urn:mpeg:mpeg21:2003:01-REL-SX-NS
mx	urn:mpeg:mpeg21:2003:01-REL-MX-NS

Insert a new subclause 3.1.3 as follows:

3.1.3 Conversion and permission-specific terms and definitions

3.1.3.1
Conversion
A process that changes the characteristics of a **resource**.

NOTE In general a conversion performs the act as defined by the ISO/IEC 21000-6 term *adapt*.

3.1.3.2
Conversion act
A **conversion** and its parameters, including the actual name of the parameters. The semantics shall be defined through ISO/IEC 21000-6.

3.1.3.3
Conversion tool
A hardware and/or software module that implements a **conversion act** in order to perform the **conversion** as specified by the ISO/IEC 21000-6 term defining this **conversion**.

Insert a new subclause 5.6 as follows:

5.6 Conversion descriptions

5.6.1 Introduction

Subclause 5.6 specifies conversion descriptions identifying conversions. Conversion descriptions and parameters can be defined within this part of ISO/IEC 21000 by referencing appropriate ISO/IEC 21000-6 terms which specify the semantics of the conversion as well as its parameters. The conversion description tool as specified within subclause 5.6 provides a basic abstract type for a subset of types that can be used for defining conversions and permissions.

5.6.2 Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ISO/IEC 21000-7, AMD/1: Conversions and Permissions -->
<!-- Schema for Conversion Descriptions -->
<schema
   version="ISO/IEC 21000-7:2004/Amd.1"
   id="ConvD-AMD1.xsd"
   targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
   elementFormDefault="qualified" attributeFormDefault="unqualified">

   <include schemaLocation="BSDLink-AMD1.xsd"/>

  <!-- ################################################  -->
  <!--  Definition of ConversionDescriptionBaseType      -->
  <!-- ################################################  -->

  <complexType name="ConversionDescriptionBaseType" abstract="true">
   <complexContent>
    <extension base="dia:DIADescriptionType">
     <sequence>
      <element name="ConversionActUri">
       <complexType>
        <attribute name="uri" type="anyURI" use="required"/>
       </complexType>
      </element>
      <element name="Parameter" type="dia:BaseParameterType"
          minOccurs="0" maxOccurs="unbounded"/>
     </sequence>
    </extension>
   </complexContent>
  </complexType>

</schema>
```

5.6.3 Semantics

Semantics of the `ConversionDescriptionBaseType`:

Name	*Definition*
`ConversionDescriptionBaseType`	`ConversionDescriptionBaseType` extends `DIADescriptionType` and provides a base abstract type for a subset of types that can be used as part of the `ConversionCapabilitiesType`, `ConversionLinkType`, `ConversionUriType` and `PermittedDiaChanges` to identify a conversion. A conversion is defined by the conversion act which references a specialized ISO/IEC 21000-6 term and its parameters.
`ConversionActUri`	Describes the conversion act using a reference to a specialized ISO/IEC 21000-6 term.
`uri`	Describes the reference to the specialized ISO/IEC 21000-6 term.
`Parameter`	Describes an input parameter to the conversion tool as defined by the specialized ISO/IEC 21000-6 term.

5.6.4 Example

EXAMPLE See subclause 6.5.29 Example 4, and subclause 14.6 for examples of conversion descriptions.

Insert the following as a new subclause 6.5.28 and increment subsequent subclauses accordingly:

6.5.28 ConversionCapabilities

6.5.28.1 Introduction

This subclause specifies the conversion capabilities of a terminal which express the types of conversions that a terminal is capable of performing.

6.5.28.2 Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ISO/IEC 21000-7, AMD/1: Conversions and Permissions -->
<!-- Schema for ConversionCapabilities -->
<schema
   version="ISO/IEC 21000-7:2004/Amd.1"
   id="ConvCapab-AMD1.xsd"
   targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
   elementFormDefault="qualified" attributeFormDefault="unqualified">

   <include schemaLocation="UCD.xsd"/>
   <include schemaLocation="UED.xsd"/>
```

```
   <include schemaLocation="ConvD-AMD1.xsd"/>

   <!-- ################################################ -->
   <!--  Definition of ConversionCapabilitiesType        -->
   <!-- ################################################ -->

   <complexType name="ConversionCapabilitiesType">
      <complexContent>
         <extension base="dia:TerminalCapabilityBaseType">
            <sequence>
               <element name="ConversionCapability"
                  type="dia:ConversionCapabilityType"
                  minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>

  <!-- ########################################### -->
  <!-- ConversionCapabilityType                    -->
  <!-- ########################################### -->

  <complexType name="ConversionCapabilityType">
   <complexContent>
    <extension base="dia:ConversionDescriptionBaseType">
     <sequence>
      <any namespace="##other" processContents="lax" minOccurs="0"/>
     </sequence>
    </extension>
   </complexContent>
  </complexType>

</schema>
```

6.5.28.3 Semantics

Semantics of the `ConversionCapabilitiesType`:

Name	*Definition*
`ConversionCapabilitiesType`	Tool for describing the conversions that a terminal is capable of performing. NOTE 1 Each child shall be a conversion capability description that identifies a conversion the terminal is capable of performing.
`ConversionCapability`	Describes a conversion that a terminal is capable of performing.

Semantics of the `ConversionCapabilityType`:

Name	*Definition*
`ConversionCapabilityType`	Tool for describing a conversion that a terminal is capable of performing.

Name	*Definition*
`any`	Describes a conversion that a terminal is capable of performing using XML. NOTE 2 Provides an extension mechanism which allows the inclusion of conversion descriptions defined within other parts of ISO/IEC 21000 or even other standardization bodies.

In subclause 6.5.29, append the following text:

EXAMPLE 4 This example describes a terminal capable of dithering and scaling.

```
<DIA>
   <Description xsi:type="TerminalsType">
      <Terminal>
         <TerminalCapability xsi:type="ConversionCapabilitiesType">
            <ConversionCapability>
               <ConversionActUri uri="urn:mpeg:mpeg21:2003:01-RDD-NS:dither"/>
            </ConversionCapability >
            <ConversionCapability>
               <ConversionActUri uri="http://www.adaptationRUs.com/scale"/>
            </ConversionCapability >
         </TerminalCapability>
      </Terminal>
   </Description>
</DIA>
```

Change the heading of clause 7 to "Conversion tools" and replace the text of the clause as follows:

7 Conversion tools

7.1 Introduction

This clause specifies syntax and semantics of the conversion tools. The conversion tools include the BSDLink and ConversionLink tools. The former provides means for linking a steering description to Bitstream Syntax Descriptions (BSDs) enabling the BSD-based adaptation approach based on the decisions provided by the steering description. The latter is complementary to the BSDLink tool and provides means for linking steering description to general-purpose conversions not covered by the BSDLink tool, e.g., transcoding or summarization.

7.2 Bitstream syntax description link

Include the ISO/IEC 21000-7:2004 text of clause 7 here, demoting each heading one level.

Replace the text of subclause 7.2.2 with the following text:

```
<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Bitstream syntax description link -->
<schema
  version="ISO/IEC 21000-7:2004/Amd.1"
```

```
id="BSDLink-AMD1.xsd"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
elementFormDefault="qualified" attributeFormDefault="unqualified">

<include schemaLocation="DIA.xsd"/>

<!-- ########################################################## -->
<!-- Definition of the BSDLinkType                              -->
<!-- ########################################################## -->

<complexType name="BSDLinkType">
  <complexContent>
    <extension base="dia:DIADescriptionType">
      <sequence>
        <element ref="dia:SteeringDescriptionRef" minOccurs="0"/>
        <element name="BSDRef" type="dia:ReferenceType"/>
        <element ref="dia:BitstreamRef" minOccurs="0">
        <element name="BSDTransformationRef" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <restriction base="anyType">
                <attribute name="uri" type="anyURI" use="required"/>
                <attribute name="type" type="anyURI" use="optional"/>
              </restriction>
            </complexContent>
          </complexType>
        </element>
        <element name="Parameter" type="dia:BaseParameterType" minOccurs="0"
           maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- ########################################################## -->
<!-- Definition of the SteeringDescriptionRef                   -->
<!-- ########################################################## -->

<element name="SteeringDescriptionRef" type="dia:ReferenceType"/>

<!-- ########################################################## -->
<!-- Definition of the BitstreamRef                             -->
<!-- ########################################################## -->

<element name="BitstreamRef">
  <complexType>
    <attribute name="uri" type="anyURI" use="required"/>
  </complexType>
</element>

<!-- ########################################################## -->
<!-- Definition of the SteeringDescriptionType                  -->
<!-- ########################################################## -->
```

```
  <complexType name="SteeringDescriptionType" abstract="true">
    <complexContent>
      <extension base="dia:DIADescriptionType"/>
    </complexContent>
  </complexType>

  <!-- ################################################## -->
  <!-- Definition of the BaseParameterType                -->
  <!-- ################################################## -->

  <complexType name="BaseParameterType" abstract="true">
    <complexContent>
      <extension base="dia:DIABaseType">
        <attribute name="name" type="QName" use="required"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- ################################################## -->
  <!-- Definition of the ConstantType                     -->
  <!-- ################################################## -->

  <complexType name="ConstantType">
    <complexContent>
      <extension base="dia:BaseParameterType">
        <sequence>
          <element name="Value" type="string"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- ################################################## -->
  <!-- Definition of the IOPinRefType                     -->
  <!-- ################################################## -->

  <complexType name="IOPinRefType">
    <complexContent>
      <extension base="dia:BaseParameterType">
        <sequence>
          <element name="Value" type="anyURI"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

</schema>
```


7.3 Conversion link

7.3.1 Introduction

This subclause specifies the syntax and semantics of the ConversionLink tool. The ConversionLink is similar to BSDLink in the way that this tool provides the facilities to link so-called steering description tools and conversion descriptions in a flexible and extensible way. However, though both tools do target conversions, the BSDLink tool targets basically the adaptation of scalable bitstreams whereas the ConversionLink tool is applicable for any kind of adaptations, e.g., transcoding, transforming, or transmoding.

The ConversionLink tool eases the referencing of information assets which can be used for conversion, i.e., references to these assets are stored in the ConversionLink. This description contains at least a reference to a conversion act which specifies the semantics of a conversion and the semantics of its generic parameters including the actual name of the parameters. Additionally, the ConversionLink tool may contain a reference to a steering description which governs the whole conversion process, a reference to the actual resource subject to the conversion and possibly a description of this resource.

A conversion decision architecture compatible with Annex B is given in Annex J.

7.3.2 ConversionLink syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Conversion link -->
<schema
   version="ISO/IEC 21000-7:2004/Amd.1"
   id="ConvL-AMD1.xsd"
   targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
   elementFormDefault="qualified" attributeFormDefault="unqualified">

   <include schemaLocation="BSDLink-AMD1.xsd"/>
   <include schemaLocation="ConvD-AMD1.xsd"/>

   <!-- ################################################   -->
   <!-- Definition of the ConversionLinkType             -->
   <!-- ################################################   -->

  <complexType name="ConversionLinkType">
   <complexContent>
    <extension base="dia:ConversionDescriptionBaseType">
     <sequence>
      <element ref="dia:SteeringDescriptionRef" minOccurs="0"/>
      <element ref="dia:BitstreamRef" minOccurs="0"/>
      <element name="ResourceDescriptionRef" minOccurs="0">
       <complexType>
        <attribute name="uri" type="anyURI" use="required"/>
       </complexType>
      </element>
     </sequence>
    </extension>
   </complexContent>
  </complexType>

  <!-- ########################################### -->
  <!-- ConversionCompositeType                     -->
  <!-- ########################################### -->

  <complexType name="ConversionCompositeType">
    <complexContent>
      <extension base="dia:DIADescriptionType">
        <sequence>
          <element name="ConversionDescription" type="dia:ConversionLinkType"
minOccurs="2" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

</schema>
```

7.3.3 ConversionLink semantics

Semantics of the `ConversionLinkType`:

Name	Definition
`ConversionLinkType`	Tool for describing suggested conversions as a link between the parameters provided by a steering description and the parameters required for a conversion.
`SteeringDescriptionRef`	Describes a reference to the steering description. The target of this reference shall be of `SteeringDescriptionType`. If this element is not present the conversion should be steered by the ISO/IEC 21000-2 choice/selection mechanism.
`ResourceDescriptionRef`	Describes a reference to a description of the resource subject to the conversion. If this element is present, this description can be used by the resource conversion tools.
`BitstreamRef`	Describes a reference to the bitstream subject to the conversion. NOTE If this element is not present the bitstream should be either provided by the application or the ISO/IEC 21000-2 `Resource` element.

Semantics of `ConversionCompositeType`:

Name	Definition
`ConversionCompositeType`	Tool for describing a conversion as a composite of other conversion links in sequence such that the conversions are given by its child elements and the order is given by the sequence of its child elements.
`ConversionDescription`	Describes a single conversion as part of the composite conversion.

7.3.4 Example

EXAMPLE The following ConversionLink example uses an AdaptationQoS description to provide the scaling factor for the conversion, which is used for spatial scaling of the JPEG image, and the required color components according to the terminal capabilities.

An Adaptation Decision-Taking Engine (ADTE) based on an AdaptationQoS description gets the values of the display resolution and the display color capabilities for the respective I/O pins and then looks up the values of the corresponding `SCALE` and `NR_COLORS` by processing the AdaptationQoS modules. The obtained values of the IOPins `SCALE` and `NR_COLORS` are then used by the ConversionLink to pass parameters to the actual image conversion tool which implements the conversion act URI.

```
<DIA>
 <Description xsi:type="ConversionLinkType">
  <ConversionActUri uri="urn:mpeg:mpeg21:2003:01-RDD-NS:ImageRescaleGrayScale"/>
  <Parameter xsi:type="IOPinRefType" name="rescale">
   <Value>SCALE</Value>
  </Parameter>
  <Parameter xsi:type="IOPinRefType" name="colors">
   <Value>NR_COLORS</Value>
  </Parameter>
  <SteeringDescriptionRef uri="#akiyo AQoS"/>
  <BitstreamRef uri="akiyo.jpg"/>
</Description>
</DIA>
```

Insert a new subclause 9.10 as follows:

9.10 CrossConversionQoS

9.10.1 Introduction

Subclause 9.10 specifies syntax and semantics of the CrossConversionQoS tool. A resource may be adapted by different methods, (referred herein as conversion options). Normally, each conversion option is associated with an AdaptationQoS description, so the adaptation engine can find an appropriate conversion level of that conversion option. When there are more than one conversion options available, the CrossConversionQoS tool helps the adaptation engine to select the best conversion option (i.e., with the highest quality) given certain constraints. The quality of a conversion option can be computed from the utilities provided by the AdaptationQoS description of that conversion option, or manually assigned by the provider. Besides elements for quality assessment, CrossConversionQos has a reference to ConversionLink or BSDLink, which provides the details of a conversion option.

9.10.2 CrossConversionQoS syntax

```
<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Cross Conversion QoS -->
<schema
   version="ISO/IEC 21000-7:2004/Amd.1"
   id="CCQoS-AMD1.xsd"
   targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
   xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
   elementFormDefault="qualified" attributeFormDefault="unqualified">

   <include schemaLocation="AQoS.xsd"/>
   <import namespace="urn:mpeg:mpeg7:schema:2001"
       schemaLocation="mpeg7-udp-2003.xsd"/>

  <!-- ################################################ -->
  <!-- Definition of CrossConversionQoSType             -->
  <!-- ################################################ -->

  <complexType name="CrossConversionQoSType">
   <complexContent>
    <extension base="dia:DIADescriptionType">
     <sequence>
      <element name="ConversionOption" type="dia:ConversionOptionType"
```

```
          minOccurs="2" maxOccurs="unbounded"/>
      </sequence>
     </extension>
    </complexContent>
   </complexType>

   <!-- ################################################ -->
   <!-- Definition of ConversionOptionType               -->
   <!-- ################################################ -->

   <complexType name="ConversionOptionType">
    <complexContent>
     <extension base="dia:DIABaseType">
      <sequence>
       <element name="Modality" type="mpeg7:ControlledTermUseType"
           minOccurs="0"/>
       <element name="LinkRef" type="dia:ReferenceType"/>
       <element name="QualityFunction">
        <complexType>
         <complexContent>
          <extension base="dia:BaseStackFunctionType">
           <attribute name="scale" type="float" use="optional" default="1"/>
          </extension>
         </complexContent>
        </complexType>
       </element>
      </sequence>
     </extension>
    </complexContent>
   </complexType>

</schema>
```

9.10.3 CrossConversionQoS semantics

Semantics of the `CrossConversionQoSType`:

Name	Definition
`CrossConversionQoSType`	Tool for describing different conversion options of a resource and their quality assessments, so as to help the adaptation engine finding the conversion which has the highest quality.
`ConversionOption`	Describes a conversion option.

Semantics of the `ConversionOptionType`:

Name	Definition
`ConversionOptionType`	Describes a conversion option and its associated quality assessment which is obtained from the `QualityFunction`.

Modality	Describes the modality or format of the converted Resource. CS's that may be used for this purpose are the ContentCS, the AudioCodingFormatCS and the VisualCodingFormatCS defined in ISO/IEC 15938-5.
LinkRef	Describes a reference to a BSDLink or ConversionLink description.
QualityFunction	Describes a function to compute the quality from various utilities. The function, which is of BaseStackFunctionType, will return a numerical value. .
scale	Describes a multiplicative factor to be applied to the QualityFunction result.

9.10.4 Example

EXAMPLE In this example, we have a video resource which can be adapted by two conversion options, one is video transforming (or scaling) by requantization and the other is video-to-slideshow conversion. Each conversion option is provided with a utility function describing the characteristics like utility, bitrate corresponding to different adaptation levels (operators). CrossConversionQoS is employed to compare this two different conversion options. The slideshow quality is defined by:

$$Slideshow_Q = Slideshow_{PSNR} * 1.9$$

The weighting factor 1.9 is assigned to the value of the scale attribute of image modality(i.e. slideshow here). The two utility functions of video and slideshow can be compared on the same scoring scale. When VIDEO_PSNR ≥ SLIDESHOW_PSNR*1.9, video conversion is selected; and when VIDEO_PSNR < SLIDESHOW_PSNR*1.9, slideshow conversion is selected.

```
<DIA>
  <DescriptionMetadata>
    <ClassificationSchemeAlias
      alias="AQoS" href="urn:mpeg:mpeg21:2003:01-DIA-AdaptationQoSCS-NS"/>
    <ClassificationSchemeAlias
      alias="SFO" href="urn:mpeg:mpeg21:2003:01-DIA-StackFunctionOperatorCS-NS"/>
    <ClassificationSchemeAlias
      alias="MEI" href="urn:mpeg:mpeg21:2003:01-DIA-MediaInformationCS-NS"/>
  </DescriptionMetadata>

  <Description xsi:type="CrossConversionQoSType">
    <ConversionOption>
      <Modality href="urn:mpeg:mpeg7:cs:ContentCS:2001:4.2">
        <mpeg7:Name>Video</mpeg7:Name>
      </Modality>
      <LinkRef uri="#Video_ConvLink"/>
      <QualityFunction>
        <Argument xsi:type="InternalIOPinRefType" iOPinRef="VIDEO_PSNR"/>
      </QualityFunction>
    </ConversionOption>
    <ConversionOption>
      <Modality href="urn:mpeg:mpeg7:cs:ContentCS:2001:4.1">
        <mpeg7:Name>Image</mpeg7:Name>
      </Modality>
      <LinkRef uri="#Slideshow_ConvLink"/>
      <QualityFunction scale="1.9">
        <Argument xsi:type="InternalIOPinRefType" iOPinRef="SLIDESHOW_PSNR"/>
      </QualityFunction>
```

```
    </ConversionOption>
  </Description>

  <!-- 1st Utility function for video modality with associated UCD and
ConversionLink -->
  <Description xsi:type="AdaptationQoSType" id="Video_AQoS">
    <Module xsi:type="UtilityFunctionType">
      <Constraint iOPinRef="VIDEO_BITRATE">
        <Values xsi:type="FloatVectorType">
          <Vector>1073 678 455 196 80 56</Vector>
        </Values>
      </Constraint>
      <AdaptationOperator iOPinRef="VIDEO_QUANT_SCALE">
        <Values xsi:type="FloatVectorType">
          <Vector>2 3 4 8 20 40</Vector>
        </Values>
      </AdaptationOperator>
      <AdaptationOperator iOPinRef="VIDEO_GOP_SIZE">
        <Values xsi:type="FloatVectorType">
          <Vector>12 12 30 50 100 200</Vector>
        </Values>
      </AdaptationOperator>
      <Utility iOPinRef="VIDEO_PSNR">
        <Values xsi:type="FloatVectorType">
          <Vector>49.16 45.35 42.88 38.78 34.05 31.78</Vector>
        </Values>
      </Utility>
    </Module>

    <IOPin semantics=":AQoS:1.1.1" id="VIDEO_BITRATE"/>
    <IOPin semantics=":AQoS:2.1" id="VIDEO_PSNR"/>
    <IOPin semantics=":AQoS:1.3.3" id="VIDEO_QUANT_SCALE"/>
    <IOPin id="VIDEO_GOP_SIZE"/>

    <!-- Reference to the UCD for Optimization -->
    <Constraints uri="#Video_UCD"/>
  </Description>

  <!-- UCD for video transforming adaptation -->
  <Description xsi:type="UCDType" id="Video_UCD">
    <AdaptationUnitConstraints>
    <!-- Constraint for video transforming : VIDEO_BITRATE <= Bandwidth UED -->
      <LimitConstraint>
        <Argument xsi:type="InternalIOPinRefType" iOPinRef="VIDEO_BITRATE"/>
        <Argument xsi:type="SemanticalRefType" semantics=":MEI:6"/>
        <Operation operator=":SFO:38"/>
      </LimitConstraint>
    </AdaptationUnitConstraints>
  </Description>

  <!-- Video transforming adaptation -->
  <Description xsi:type="ConversionLinkType" id="Video_ConvLink">
    <ConversionActUri uri="urn:enst:videotransforming123"/>
    <Parameter xsi:type="IOPinRefType" name="qscale">
      <Value>VIDEO_QUANT_SCALE</Value>
    </Parameter>
    <Parameter xsi:type="IOPinRefType" name="gop_size">
      <Value>VIDEO_GOP_SIZE</Value>
    </Parameter>
    <SteeringDescriptionRef uri="#Video_AQoS"/>
    <BitstreamRef uri="mpeg4_video.mp4"/>
```

```
  </Description>

  <!-- 2st utility function for image modality with associated UCD and
ConversionLink -->
  <Description xsi:type="AdaptationQoSType" id="Slideshow_AQoS">
    <Module xsi:type="UtilityFunctionType">
      <Constraint iOPinRef="SLIDESHOW_BITRATE">
        <Values xsi:type="FloatVectorType">
          <Vector>106 64 32</Vector>
        </Values>
      </Constraint>
      <AdaptationOperator iOPinRef="SLIDESHOW_PERIOD">
        <Values xsi:type="IntegerVectorType">
          <Vector>3 5 10</Vector>
        </Values>
      </AdaptationOperator>
      <Utility iOPinRef="SLIDESHOW_PSNR">
        <Values xsi:type="FloatVectorType">
          <Vector>18.7 17.3 16.8</Vector>
        </Values>
      </Utility>
    </Module>

    <IOPin semantics=":AQoS:1.1.1" id="SLIDESHOW_BITRATE"/>
    <IOPin semantics=":AQoS:2.1" id="SLIDESHOW_PSNR"/>
    <IOPin id="SLIDESHOW_PERIOD"/>

    <!-- Reference to the UCD for Optimization -->
    <Constraints uri="#Slideshow_UCD"/>
  </Description>

  <!-- UCD for video to Slideshow conversion -->
  <Description xsi:type="UCDType" id="Slideshow_UCD">
    <AdaptationUnitConstraints>
    <!-- Constraint for summarization : SLIDESHOW_BITRATE <= Bandwidth UED -->
      <LimitConstraint>
        <Argument xsi:type="InternalIOPinRefType" iOPinRef="SLIDESHOW_BITRATE"/>
        <Argument xsi:type="SemanticalRefType" semantics=":MEI:6"/>
        <Operation operator=":SFO:38"/>
      </LimitConstraint>
    </AdaptationUnitConstraints>
  </Description>

  <!-- Video to Slideshow conversion -->
  <Description xsi:type="ConversionLinkType" id="Slideshow_ConvLink">
    <ConversionActUri uri="urn:enst:videosummarization123"/>
    <Parameter xsi:type="IOPinRefType" name="period">
      <Value>SLIDESHOW_PERIOD</Value>
    </Parameter>
    <SteeringDescriptionRef uri="#Slideshow_AQoS"/>
    <BitstreamRef uri="mpeg4_video.mp4"/>
  </Description>
</DIA>
```

Insert a new clause 14 as follows:

14 Change Conditions

14.1 Introduction

ISO/IEC 21000-5 and ISO/IEC 21000-6 provide the tools to permit playing, modifying, and adapting by controlling the kinds of things that can be changed. This part of ISO/IEC 21000 provides tools for use with ISO/IEC 21000-5 to provide the means by which that control over the changes that can occur when playing, modifying, or adapting digital items and their component resources can be effected.

14.2 Additional terms, definitions, symbols, abbreviated terms, namespaces, and conventions for this clause

The terms, definitions, symbols, and abbreviated terms given in clause 3 of ISO/IEC 21000-5 and namespaces and conventions given in clause 4 of ISO/IEC 21000-5 shall additionally apply to this clause.

14.3 Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ISO/IEC 21000-7, AMD/1: Conversions and Permissions -->
<!-- Schema for Change Conditions -->
<schema
   version="ISO/IEC 21000-7:2004/Amd.1"
   id="ChangeCond-AMD1.xsd"
   targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
   xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
   elementFormDefault="qualified" attributeFormDefault="unqualified">

   <import namespace="urn:mpeg:mpeg21:2003:01-REL-R-NS"
      schemaLocation="rel-r.xsd"/>
   <include schemaLocation="UCD.xsd"/>
   <include schemaLocation="UED.xsd"/>
   <include schemaLocation="ConvD-AMD1.xsd"/>

   <!-- ################################################ -->
   <!--  Definition of PermittedDiaChanges               -->
   <!-- ################################################ -->

   <complexType name="PermittedDiaChanges">
      <complexContent>
         <extension base="r:Condition">
            <sequence>
               <element name="ConversionDescription"
                  type="dia:ConversionDescriptionBaseType"
                  minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>

   <!-- ################################################ -->
   <!--  Definition of ChangeConstraint                  -->
   <!-- ################################################ -->
```

```
   <complexType name="ChangeConstraint">
      <complexContent>
         <extension base="r:Condition">
            <sequence minOccurs="0">
               <element name="steeringInput" type="r:Resource"
                  minOccurs="0"/>
               <element name="constraint" type="dia:UCDType"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>

   <!-- ################################################ -->
   <!--  Definition of permittedDiaChanges               -->
   <!-- ################################################ -->

   <element name="permittedDiaChanges" type="dia:PermittedDiaChanges"
      substitutionGroup="r:condition"/>

   <!-- ################################################ -->
   <!--  Definition of changeConstraint                  -->
   <!-- ################################################ -->

   <element name="changeConstraint" type="dia:ChangeConstraint"
      substitutionGroup="r:condition"/>

   <!-- ########################################### -->
   <!-- ConversionUriType                           -->
   <!-- ########################################### -->

   <complexType name="ConversionUriType">
    <complexContent>
     <extension base="dia:ConversionDescriptionBaseType"/>
    </complexContent>
   </complexType>

</schema>
```

14.4 Authorization Context Properties

Property name	Property value	Statement represented
dia:diaChanges()	*d*	*d* is a `dia:PermittedDiaChanges`, each child of *d* identifies one conversion being used for the formal or format changes (according to the semantics of the Right Member of the authorization request) during the requested performance, and each conversion being used for the formal or format changes (according to the semantics of the Right Member of the authorization request) during the requested performance is identified by some child of *d*. NOTE Some conversion descriptions are given in subclause 5.6.
dia:ucd(*d*)	true	*d* is a `dia:UCDType` and there is at least one possible output of a UCD-based ADTE taking as inputs a null steering description, UCD *d*, and accurate resolvers such that the output result code is success and the values of variables in the output solution (when interpreted according to the semantics ascribed to the variables) accurately describes the changes (according to the semantics of the Right Member of the authorization request) occurring during the requested performance.

dia:ucdt(*t*, *d*)	true	*t* is an `r:Resource` identifying a steering description, *d* is a `dia:UCDType`, and there is at least one possible output of a UCD-based adaptation decision-taking engine taking as inputs steering description *t*, UCD *d*, and accurate resolvers such that the output result code is success and the values of variables in the output solution (when interpreted according to the semantics ascribed to the variables) accurately describes the changes (according to the semantics of the Right Member of the authorization request) occurring during the requested performance.

NOTE Note that authorization context property names do not have any inherent relationship to elements and types defined in the schema. The relation of elements and types defined in the schema to authorization context properties is given in the semantics of those elements and types. For reading convenience names for authorization context properties are chosen to be similar to the elements and types whose semantics reference them.

14.5 Semantics

Semantics of the `PermittedDiaChanges`:

Let *c* be a `dia:PermittedDiaChanges`. Let (*p*, *r*, *t*, *v*, *Σ*, *L*, *R*) be an authorization request. Let (*g*, *h*, *e*) be an authorization story. Then *c* is Satisfied with respect to (*p*, *r*, *t*, *v*, *Σ*, *L*, *R*) and (*g*, *h*, *e*) if and only if each child element of *Σ*.dia:diaChanges() is Equal to at least one of the children of *c*.

Semantics of the `ChangeConstraint`:

Let *c* be an `dia:ChangeConstraint`. Let (*p*, *r*, *t*, *v*, *Σ*, *L*, *R*) be an authorization request. Let (*g*, *h*, *e*) be an authorization story. Then *c* is Satisfied with respect to (*p*, *r*, *t*, *v*, *Σ*, *L*, *R*) and (*g*, *h*, *e*) if and only if either *c*/`dia:steeringInput` is absent and *Σ*.dia:ucd(*c*/`dia:constraint`) is true or *c*/`dia:steeringInput` is present and *Σ*.dia:ucdt(*c*/`dia:steeringInput`, *c*/`dia:constraint`) is true.

Semantics of the `ConversionUriType`:

Name	Definition
ConversionUriType	Tool for describing a conversion using a URI which references a specialized ISO/IEC 21000-6 term and its parameters.

14.6 Example

EXAMPLE The following example shows a License to express that Anthony says that a video file can be played by Barney provided that:

- the only formal change Barney makes (if any) is spatial resolution reduction,
- the played aspect ratio is the same as the original file aspect ratio, and
- the played spatial resolution is less than 352x240.

```
<r:license>

   <r:inventory>
      <!-- ... -->
   </r:inventory>

   <r:grant>
```

```
      <!-- Barney may play the video ... -->
      <r:keyHolder licensePartIdRef="Barney"/>
      <mx:play/>
      <mx:diReference licensePartIdRef="video"/>

      <!-- ... under these conditions ... -->
      <r:allConditions>

         <!-- He is only permitted to do spatial size reduction.
              He may use Acme's algorithm (not registered in RDD) or
              he may use any algorithm registered in RDD as a SizeChange.
              -->
         <dia:permittedDiaChanges>
            <dia:ConversionDescription xsi:type="dia:ConversionUriType">
               <dia:ConversionActUri uri="urn:acme:spatialSizeReduction"/>
            </dia:ConversionDescription>
            <dia:ConversionDescription xsi:type="dia:ConversionUriType">
               <dia:ConversionActUri uri="urn:mpeg:mpeg21:2003:01-RDD-
NS:SizeChange"/>
            </dia:ConversionDescription>
         </dia:permittedDiaChanges>

         <!-- These constraints apply whether or not spatial size is reduced. -->
         <dia:changeConstraint>
            <dia:constraint>
               <dia:AdaptationUnitConstraints>

                  <!-- Width must be less than 352. -->
                  <dia:LimitConstraint>
                     <dia:Argument xsi:type="dia:SemanticalRefType" semantics=
                        "urn:mpeg:mpeg21:2003:01-DIA-MediaInformationCS-NS:17"
                     />
                     <dia:Argument xsi:type="dia:ConstantDataType">
                        <dia:Constant xsi:type="dia:IntegerType">
                           <dia:Value>352</dia:Value>
                        </dia:Constant>
                     </dia:Argument>
                     <dia:Operation operator=
                        "urn:mpeg:mpeg21:2003:01-DIA-StackFunctionOperatorCS-
NS:12"/>
                  </dia:LimitConstraint>

                  <!-- Height must be less than 240. -->
                  <dia:LimitConstraint>
                     <dia:Argument xsi:type="dia:SemanticalRefType" semantics=
                        "urn:mpeg:mpeg21:2003:01-DIA-MediaInformationCS-NS:18"
                     />
                     <dia:Argument xsi:type="dia:ConstantDataType">
                        <dia:Constant xsi:type="dia:IntegerType">
                           <dia:Value>240</dia:Value>
                        </dia:Constant>
                     </dia:Argument>
                     <dia:Operation operator=
                        "urn:mpeg:mpeg21:2003:01-DIA-StackFunctionOperatorCS-
NS:12"/>
                  </dia:LimitConstraint>

                  <!-- Final aspect ratio must equal input aspect ratio. -->
                  <dia:LimitConstraint>
                     <dia:Argument xsi:type="dia:SemanticalRefType" semantics=
```

```
                        "urn:mpeg:mpeg21:2003:01-DIA-MediaInformationCS-NS:19"/>
                     <dia:Argument xsi:type="dia:SemanticalDataRefType"
                        semantics=
                        "urn:mpeg:mpeg21:2003:01-DIA-MediaInformationCS-NS:19"/>
                     <dia:Operation operator=
                        "urn:mpeg:mpeg21:2003:01-DIA-StackFunctionOperatorCS-
NS:11"/>
                  </dia:LimitConstraint>

               </dia:AdaptationUnitConstraints>
            </dia:constraint>
         </dia:changeConstraint>

      </r:allConditions>

   </r:grant>

   <r:issuer>
      <r:keyHolder licensePartIdRef="Anthony"/>
   </r:issuer>

</r:license>
```

Insert a new informative Annex H as follows:

Annex H
(informative)

Conversion Description Examples

H.1 Introduction

This Annex describes how a conversion description for the cropping of a rectangular bitmap image can be created in an interoperable way (leveraging ISO/IEC 21000-6) and used within the conversion description framework. These steps are followed:

1. Elaborate the semantics of cropping a rectangular bitmap image.
2. Identify existing RDD terms related to those semantics.
3. Specialize existing RDD terms to arrive at a CropRectangularBitmapImage term.
4. Write the syntax and semantics for any relevant parameters needed to describe the conversion.
5. Write the conversion description.
6. Use the conversion description in the conversion description framework.

It is important to note that these steps only describe how to use a conversion *description* within the conversion description framework. They do not go beyond that and describe how to use a *conversion*. The methods of using a conversion go beyond the scope of this Annex and beyond the scope of this Part of ISO/IEC 21000.

In order to perform a conversion, a party might be required to have authorization from some other party. This authorization can be conveyed to that first party by that other party using a description of that conversion in a rights expression (see ISO/IEC 21000-5) according to the conversion description framework (see specifically clause 14).

H.2 Elaborate the semantics of cropping a rectangular bitmap image

Cropping refers to spatial cropping. The following statements provide more details about what is meant by cropping a rectangular bitmap image:

- Temporal cropping is not considered.
- Bitmap images consist of a two-dimensional array of pixel values. Therefore, decoding a bitmap into pixel values and re-encoding those pixel values into a bitmap is lossless.
- No pixel value is affected by its adjacent pixel values, and removing any pixel has no effect on the remaining pixels.
- Pixels are arranged in a rectangular grid (a grid with right angles), so any two pixel locations (or a pixel location and a width and height) are sufficient to describe a rectangular block of pixels that clearly include or exclude pixels from the original image (no pixel is half included and half excluded from the cropped region).
- The following mathematical description provide more details about what is meant by cropping a rectangular bitmap image:
- Let x_in be the number of pixels along the x axis of the input resource.
- Let y_in be the number of pixels along the y axis of the input resource.
- Let x_out be the number of pixels along the x axis of the output resource.
- Let y_out be the number of pixels along the y axis of the output resource.
- Let pixels_in[x][y] be a zero-based array representation of the pixels in the input resource.
- Let pixels_out[x][y] be a zero-based array representation of the pixels in the output resource.
- Let x_offset be some integer representing the x coordinate in the input resource of the upper-left corner of the output resource.
- Let y_offset be some integer representing the y coordinate in the input resource of the lower-right corner of the output resource.
- Then the following constraints must hold:
  - 0 <= x_offset < x_in
  - 0 <= y_offset < y_in
  - 0 < x_out <= x_in
  - 0 < y_out <= y_in
  - x_offset + x_out <= x_in
  - y_offset + y_out <= y_in
  - for each x where 0 <= x < x_out, for each y where 0 <= y < y_out, pixels_out[x][y] = pixels_in[x+x_offset][y+y_offset]

H.3 Identify existing RDD terms related to those semantics

The concept of a rectangular bitmap image relates to the existing RDD term Resource.

The concepts of x, y, x_in, y_in, x_out, y_out, x_offset, and y_offset relate to the existing RDD term Quantity.

The concept of number of pixels relates to the existing RDD term UnitOfMeasure.

The concept of numerical pixel value (pixels_in[x][y] and pixels_out[x][y]) relates to the existing RDD term Quantity.

The concept of [x] and [y] coordinates indicating a location in a bitmap relates to the existing RDD term Place.

The concept of cropping relates to the existing RDD term Diminish, defined in the RDD as:

> "To Derive a new Resource which is smaller than its Source."

In the RDD Standard the (multiple) inheritance hierarchy for the Act "Diminish" is:

Act
- Do
--- Make
----- Adapt
------- **Diminish**
--- InteractWith
----- Change
------- ChangeTransiently
--------- Adapt
---------- **Diminish**

Its related family of Terms (eg Diminisher, Diminution, SourceOfDiminution, TimeOfDiminishing etc) all follow parallel hierarchies.

The scope of Diminish is defined in RDD and REL as follows:

> "With *Diminish*, two distinct Resources will exist at the end of the process, one of which [SourceOfDiminution] is the original Resource in unchanged form, and one of which [Diminution] is newly made, whose content is Adapted from the original Resource, and a Measure of which is smaller than that of the original while no Measures of it are larger. Changes can include the removal of elements of the original Resource. Changes can be made temporarily to the original Resource in the course of the Diminish process, but such changes are not saved in the original Resource at the end of the process."

Using the representation syntax[1] of the RDD standard, the elements of a ContextView for a DiminishingEvent are as follows:

#1[DiminishingEvent] icoAgent #2.n[Diminisher][occ:1-n]
#1[DiminishingEvent] icoResource #3.n[Diminution][occ:1-n]
#1[DiminishingEvent] icoResource #4.n[SourceOfDiminution][occ:1-n]
#1[DiminishingEvent] icoTime #5.n[TimeOfDiminishing][occ:1-n]
#1[DiminishingEvent] icoPlace #6.n[PlaceOfDiminishing][occ:1-n]
#1[DiminishingEvent] icoPlace #7.n[PlaceOfDiminishingFrom][occ:1-n]
#7.n IsPartOf #6.n
#7.n IsPlaceOf #4.n
#1[DiminishingEvent] icoPlace #8.n[PlaceOfDiminishingTo][occ:1-n]
#8.n IsPartOf #6.n

[1] The element #n represents a value of an instance of the bracketed class in each case (for example, #1[DiminishingEvent] represents an instance of a DiminishingEvent). The form #n.n represents each instance of the bracketed class in the Context, though that has no impact in this example. The abbreviation "ico" stands for "IsContextOf".

#8.n IsEquivalentTo #7.n [ver:Possible]
#8.n IsPlaceOf #3.n

For ease of reading this can be serialized to these simpler, nested RDD triples:

#1 IsA DiminishingEvent
#1 icoAgent #2.n [occ:1-n]
    #2.n IsA Diminisher
#1 icoResource #3.n [occ:1-n]
    #3.n IsA Diminution
#1 icoResource #4.n [occ:1-n]
    #4.n IsA SourceOfDiminution
#1 icoTime #5.n [occ:1-n]
    #5.n IsA TimeOfDiminishing
#1 icoPlace #6.n [occ:1-n]
    #6.n IsA PlaceOfDiminishing
#1 icoPlace #7.n [occ:1-n]
    #7.n IsA PlaceOfDiminishingFrom
    #7.n IsPartOf #6.n
    #7.n IsPlaceOf #4.n
#1 icoPlace #8.n [occ:1-n]
    #8.n IsA PlaceOfDiminishingTo
    #8.n IsPartOf #6.n
    #8.n IsEquivalentTo #7.n [ver:Possible]
    #8.n IsPlaceOf #3.n

H.4 Specialize existing RDD terms to arrive at a CropRectangularBitmapImage term

The process of specialization is to add the necessary additional constraints to elements of the ContextView.

The requirements add no specialized constraints on the Agent, Time or Place, so all specialization is confined to the attributes of the Resources: the input (SourceOfDiminution) and the (Diminution). The triples to be specialized are therefore:

#1 icoResource #3.n [occ:1-n]
    #3.n IsA Diminution
#1 icoResource #4.n [occ:1-n]
    #4.n IsA SourceOfDiminution

This Act relates to a single input and a single output:

#1 icoResource **#3** [occ:**1**]
    #3 IsA Diminution
#1 icoResource **#4** [occ:**1**]
    #4 IsA SourceOfDiminution

Each resource is a BitmapImage (BitmapImage needs to be defined as an RDD Term):

#1 icoResource #3 [occ:1]
    #3 IsA Diminution
    #3 IsA BitmapImage
#1 icoResource #4 [occ:1]
    #4 IsA SourceOfDiminution
    #4 IsA BitmapImage

Each resource is Rectangular (Rectangular must be defined as an RDD Term). This might be shown thus:

#1 icoResource #3 [occ:1]
#3 IsA Diminution
#3 IsA BitmapImage
#3 Is Rectangular
#1 icoResource #4 [occ:1]
#4 IsA SourceOfDiminution
#4 IsA BitmapImage
#4 Is Rectangular

or as this is a common resource type a new subtype of BitmapImage might be defined according to the following heirarchy:

Resource
- **BitmapImage**
--- **RectangularBitmapImage**

and used thus:

#1 icoResource #3 [occ:1]
#3 IsA Diminution
#3 IsA **Rectangular**BitmapImage
#1 icoResource #4 [occ:1]
#4 IsA SourceOfDiminution
#4 IsA **Rectangular**BitmapImage

BitmapHeight, BitmapWidth, and NumberOfPixels would need to be defined as RDD Terms. Such definition might result in the following heirarchy:

Quantity
- **Distance**
--- **Height**
----- **BitmapHeight**
--- **Width**
----- **BitmapWidth**

UnitOfMeasure
- **Count**
--- **NumberOfPixels**

Each image has a vertical and horizontal axis measured in NumberOfPixels:

#1 icoResource #3 [occ:1]
#3 IsA Diminution
#3 IsA RectangularBitmapImage
#3 HasQuantity #9
#9 IsA BitmapHeight
#9 HasUnitOfMeasure NumberOfPixels
#3 HasQuantity #10
#10 IsA BitmapWidth
#10 HasUnitOfMeasure NumberOfPixels
#1 icoResource #4 [occ:1]
#4 IsA SourceOfDiminution
#4 IsA RectangularBitmapImage
#4 HasQuantity #11
#11 IsA BitmapHeight
#11 HasUnitOfMeasure NumberOfPixels

#4 HasQuantity #12

#12 IsA BitmapWidth

#12 HasUnitOfMeasure NumberOfPixels

The “HasQuantity” may instead be specialized itself to “HasBitmapHeight” and “HasBitmapWidth”, but the elements still require definition.

Some mathematical functions are now needed. RDD should Adopt its semantics for mathematical functions from elsewhere (eg MML), but for illustration some of these might be expressed along these lines.

First, the comparison of the Heights and Widths of the inputs and outputs:

#1 icoResource #3 [occ:1]
- #3 IsA Diminution
- #3 IsA RectangularBitmapImage
- #3 HasQuantity #9
  - #9 IsA BitmapHeight
  - #9 HasUnitOfMeasure NumberOfPixels
  - **#9 IsLessThanOrEqualTo #11**
- #3 HasQuantity #10
  - #10 IsA BitmapWidth
  - #10 HasUnitOfMeasure NumberOfPixels
  - **#10 IsLessThanOrEqualTo #12**

#1 icoResource #4 [occ:1]
- #4 IsA SourceOfDiminution
- #4 IsA RectangularBitmapImage
- #4 HasQuantity #11
  - #11 IsA BitmapHeight
  - #11 HasUnitOfMeasure NumberOfPixels
- #4 HasQuantity #12
  - #12 IsA BitmapWidth
  - #12 HasUnitOfMeasure NumberOfPixels

CropWidthOffset and CropHeightOffset need to be defined as RDD Terms. Such definition might result in the following heirarchy:

Quantity
- Distance
--- Height
----- BitmapHeight
----- **CropHeightOffset**
--- Width
----- BitmapWidth
----- **CropWidthOffset**

The positioning of the output image in relation to the source is described by the relationship of the output co-ordinates to that of the source:

#1 icoResource #3 [occ:1]
- #3 IsA Diminution
- #3 IsA RectangularBitmapImage
- #3 HasQuantity #9
  - #9 IsA BitmapHeight
  - #9 HasUnitOfMeasure NumberOfPixels
  - #9 IsLessThanOrEqualTo #11
- #3 HasQuantity #10

#10 IsA BitmapWidth
#10 HasUnitOfMeasure NumberOfPixels
#10 IsLessThanOrEqualTo #12
#3 HasQuantity #13
#13 IsA CropHeightOffset
#13 HasUnitOfMeasure NumberOfPixels
#13 IsGreaterThanOrEqualTo "0"
#13 IsLessThan #11
#3 HasQuantity #14
#14 IsA CropWidthOffset
#14 HasUnitOfMeasure NumberOfPixels
#14 IsGreaterThanOrEqualTo "0"
#14 IsLessThan #12
#1 icoResource #4 [occ:1]
#4 IsA SourceOfDiminution
#4 IsA RectangularBitmapImage
#4 HasQuantity #11
#11 IsA BitmapHeight
#11 HasUnitOfMeasure NumberOfPixels
#4 HasQuantity #12
#12 IsA BitmapWidth
#12 HasUnitOfMeasure NumberOfPixels

NumericalPixelValue, PixelXCoordinate, PixelYCoordinate, and PixelPlanarCoordinateLocation would need to be defined as RDD Terms. Such definition might result in the following heirarchy:

Place
- **PixelPlanarCoordinateLocation**

Quantity
- Distance
--- Height
----- BitmapHeight
----- CropHeightOffset
----- **PixelYCoordinate**
--- Width
----- BitmapWidth
----- CropWidthOffset
----- **PixelXCoordinate**
- **ColorQuantity**
--- **NumericalPixelValue**

In order to express the relation between the source and destination pixel locations and crop offset, a further mathematical function to represent a sum (suggested here as **IsSumOf**) and RDD notation to represent a list (suggested here as **{#1, #2}**) of values to sum is needed.

Finally, it needs to be declared that the numerical pixel value of each pixel in the output is numerically equal to the corresponding numerical pixel value in the input:

#1 icoResource #3 [occ:1]
#3 IsA Diminution
#3 IsA RectangularBitmapImage
#3 HasQuantity #9
#9 IsA BitmapHeight
#9 HasUnitOfMeasure NumberOfPixels
#9 IsLessThanOrEqualTo #11

#3 HasQuantity #10
#10 IsA BitmapWidth
#10 HasUnitOfMeasure NumberOfPixels
#10 IsLessThanOrEqualTo #12
#3 HasQuantity #13
#13 IsA CropHeightOffset
#13 HasUnitOfMeasure NumberOfPixels
#13 IsGreaterThanOrEqualTo “0”
#13 IsLessThan #11
#3 HasQuantity #14
#14 IsA CropWidthOffset
#14 HasUnitOfMeasure NumberOfPixels
#14 IsGreaterThanOrEqualTo “0”
#14 IsLessThan #12
#3 HasPart #15.n [occ:1-n]
#15.n IsA Pixel
#15.n HasNumericalPixelValue #17.n
#15.n HasPlace #18.n
#18.n IsA PlanarCoordinateLocation
#18.n HasYCoordinate #19.n
#18.n HasXCoordinate #20.n
#1 icoResource #4 [occ:1]
#4 IsA SourceOfDiminution
#4 IsA RectangularBitmapImage
#4 HasQuantity #11
#11 IsA BitmapHeight
#11 HasUnitOfMeasure NumberOfPixels
#4 HasQuantity #12
#12 IsA BitmapWidth
#12 HasUnitOfMeasure NumberOfPixels
#4 HasPart #16.n [occ:1-n]
#16.n IsA Pixel
#16.n HasNumericalPixelValue #17.n
#16.n HasPlace #21.n
#21.n IsA PlanarCoordinateLocation
#21.n HasYCoordinate #22.n
#22.n IsSumOf {#13, #19.n}
#21.n HasXCoordinate #23.n
#23.n IsSumOf {#14, #20.n}

So, the ContextView for CropRectangularBitmapImage might be a specialization of a DiminishingEvent looking like this:

#1 IsA DiminishingEvent
#1 icoAgent #2.n [occ:1-n]
#2.n IsA Diminisher
#1 icoResource #3 [occ:1]
#3 IsA Diminution
#3 IsA RectangularBitmapImage

#3 HasQuantity #9
#9 IsA BitmapHeight
#9 HasUnitOfMeasure NumberOfPixels
#9 IsLessThanOrEqualTo #11
#3 HasQuantity #10
#10 IsA BitmapWidth
#10 HasUnitOfMeasure NumberOfPixels
#10 IsLessThanOrEqualTo #12
#3 HasQuantity #13
#13 IsA CropHeightOffset
#13 HasUnitOfMeasure NumberOfPixels
#13 IsGreaterThanOrEqualTo “0”
#13 IsLessThan #11
#3 HasQuantity #14
#14 IsA CropWidthOffset
#14 HasUnitOfMeasure NumberOfPixels
#14 IsGreaterThanOrEqualTo “0”
#14 IsLessThan #12
#3 HasPart #15.n [occ:1-n]
#15.n IsA Pixel
#15.n HasNumericalPixelValue #17.n
#15.n HasPlace #18.n
#18.n IsA PlanarCoordinateLocation
#18.n HasYCoordinate #19.n
#18.n HasXCoordinate #20.n
#1 icoResource #4 [occ:1]
#4 IsA SourceOfDiminution
#4 IsA RectangularBitmapImage
#4 HasQuantity #11
#11 IsA BitmapHeight
#11 HasUnitOfMeasure NumberOfPixels
#4 HasQuantity #12
#12 IsA BitmapWidth
#12 HasUnitOfMeasure NumberOfPixels
#4 HasPart #16.n [occ:1-n]
#16.n IsA Pixel
#16.n HasNumericalPixelValue #17.n
#16.n HasPlace #21.n
#21.n IsA PlanarCoordinateLocation
#21.n HasYCoordinate #22.n
#22.n IsSumOf {#13, #19.n}
#21.n HasXCoordinate #23.n
#23.n IsSumOf {#14, #20.n}
#1 icoTime #5.n [occ:1-n]
#5.n IsA TimeOfDiminishing
#1 icoPlace #6.n [occ:1-n]
#6.n IsA PlaceOfDiminishing
#1 icoPlace #7.n [occ:1-n]

#7.n IsPartOf #6.n
#7.n IsPlaceOf #4.n
#7.n IsA PlaceOfDiminishingFrom
#1 icoPlace #8.n [occ:1-n]
#8.n IsA PlaceOfDiminishingTo
#8.n IsPartOf #6.n
#8.n IsEquivalentTo #7.n [ver:Possible]
#8.n IsPlaceOf #3.n

The above definition of CropRectangularBitmapImage might result in a heirarchy like this:

Act
- Do
--- Make
----- Adapt
------- Diminish
--- InteractWith
----- Change
------- ChangeTransiently
--------- Adapt
----------- Diminish
------------- **DiminishBitmapImage**
--------------- **CropBitmapImage**
----------------- **CropRectangularBitmapImage**

Finally a point about multiple inheritance: the above tree of ActTypes is only one of many routes by which CropRectangularBitmapImage may be specialized, as RDD has to be a “lattice” rather than a “tree”. Here is another route following from Adapt rather than Diminish:

Act
- Do
--- Make
----- Adapt
--- InteractWith
----- Change
------- ChangeTransiently
--------- Adapt
----------- **AdaptBitmapImage**
------------- **AdaptRectangularBitmapImage**
--------------- CropRectangularBitmapImage

H.5 Write the conversion description

The following example describes a cropping of a rectangular bitmap image that results in a 40-by-40 rectangular bitmap image matching the upper left corner of the original image.

```
<DIA>
   <Description xsi:type="ConversionUriType">
      <ConversionActUri uri=
         "urn:mpeg:mpeg21:2003:01-RDD-NS:CropRectangularBitmapImage"/>
      <Parameter xsi:type="ConstantType" name="DiminutionBitmapHeight">
         <Value>40</Value>
      </Parameter>
      <Parameter xsi:type="ConstantType" name="DiminutionBitmapWidth">
         <Value>40</Value>
      </Parameter>
      <Parameter xsi:type="ConstantType" name="DiminutionCropHeightOffset">
         <Value>0</Value>
      </Parameter>
```

```
      <Parameter xsi:type="ConstantType" name="DiminutionCropWidthOffset">
         <Value>0</Value>
      </Parameter>
   </Description>
</DIA>
```

H.6 Use the conversion description in the conversion description framework

The conversion description can then be used inside a `dia:ConversionCapabilitiesType`, a `dia:ConversionLinkType`, a `dia:ConversionUriType`, or a `dia:PermittedDiaChanges`.

Insert a new informative Annex I as follows:

Annex I
(informative)

OWL-S Conversion Description Example

I.1 Introduction

OWL-based Web Service Ontology (OWL-S) [19] provides a computer-interpretable description of services, and the means by which they can be accessed. The structuring of the OWL-S description is motivated by the need to provide three essential types of knowledge about a service (shown in Figure I.1), each characterized by the question it answers:

— What does the service require from the user(s) or other agents as well as what does it provide for them? The answer to this question is given in the service profile. Thus, the class *Service* presents a *ServiceProfile*.

— How does it work? The answer to this question is given in the service model. Thus, the class *Service* is described by a *ServiceModel*.

— How is it used? The answer to this question is given in the service grounding. Thus, the class *Service* supports a *ServiceGrounding*.

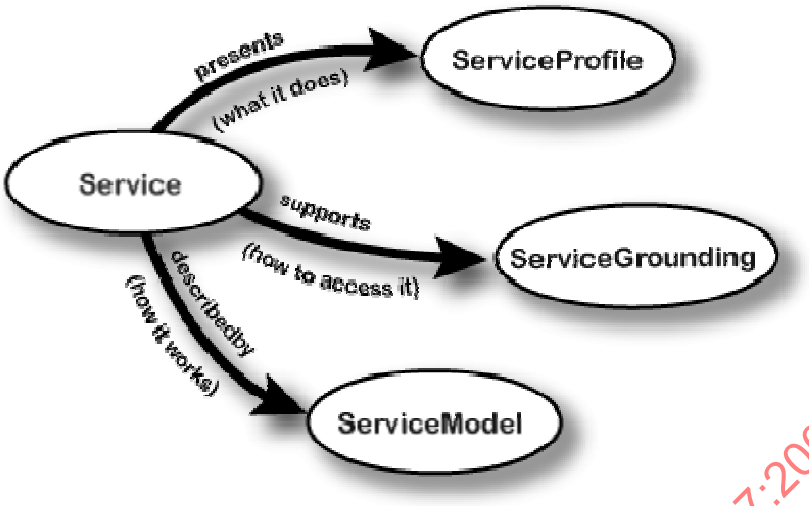


Figure I.1 — OWL-S overview [19].

I.2 Service profile

The service profile of a service provides a concise description of the service to a registry, but once the service has been selected the service profile is useless; rather, the client will use the service model to control the interaction with the service. Although the service profile and the service model play different roles during the transaction between Web services, they are two different representations of the same service. Therefore, it is natural to expect that the inputs, outputs, preconditions, and effects (IOPEs) of the service profile are reflected in the service model and vice versa.

An essential component of the service profile is the specification of what functionality the service provides and the specification of the conditions that must be satisfied for a successful result. Additionally, the service profile specifies what conditions result from the service, including the expected and unexpected results of the service activity. The OWL-S service profile represents two aspects of the functionality of the service: the information transformation (represented by inputs and outputs) and the state change produced by the execution of the service (represented by preconditions and effects). For example, a spatial scaling service for images may require the original image as input as well as the desired width and height of the image after scaling. The effect of applying the service is a spatially scaled image.

The service profile ontology does not provide a schema to describe IOPE instances. However, such a schema exists in the service model ontology, as discussed in the next section. Ideally, the IOPEs published by the service profile are a subset of those published by the service model. Therefore, the service model part of a description will create all the IOPE instances and the service profile instance can simply point to these instances.

I.3 Service model

A service model can be seen as a more fine grained description of a service which tells the user how to interact with the service. For example, a video-to-slide show service may consist of certain operations such as identifying proper images, transcoding the format, etc.

OWL-S supports three types of processes:

— Atomic processes are directly invocable. Atomic processes have no subprocesses and execute in a one step.

— Simple processes provide an abstraction mechanism to provide multiple views of the same process. Simple processes are not invocable.

— Composite processes are decomposable into other (non-composite or composite) processes.

A process can have any number of inputs (including zero), representing the information that is, under some conditions, required for the execution of the process. It can have any number of outputs, i.e., the information that the process provides. There can be any number of preconditions, which must all hold in order for the process to be invoked. Finally, the process can have any number of effects.

Inputs and outputs are described by the parameter class which carries the syntactical information about parameters: name (ID) and data type. A process cannot be invoked unless its preconditions are true. If it is invoked, it has various effects. Furthermore, preconditions and effects are represented as logical formulae.

I.4 Service grounding

I.4.1 Introduction

A service grounding (or just grounding) specifies the details of how an agent can access a service. Typically, grounding will specify a communication protocol, message formats, and other service-specific details such as port numbers used in contacting the service. Generally speaking, the service profile provides the information needed for an agent to discover a service whereas the service model and service grounding objects associated with a service provide enough information for an agent to make use of a service.

I.4.2 Web services description language

A WSDL document defines services as collections of network endpoints, or ports. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions: messages, which are abstract descriptions of the data being exchanged, and port types which are abstract collections of operations. The concrete protocol and data format specifications for a particular port type constitute a reusable binding. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service. Hence, a WSDL document uses the following elements in the definition of network services:

— Types: a container for data type definitions using some type system (such as XSD).

— Message: an abstract, typed definition of the data being communicated. Operation: an abstract description of an action supported by the service.

— Port Type: an abstract set of operations supported by one or more endpoints.

— Binding: a concrete protocol and data format specification for a particular port type.

— Port: a single endpoint defined as a combination of a binding and a network address.

— Service: a collection of related endpoints.

I.5 Example

I.5.1 Introduction

The following example provides the service profile, the service model, the service grounding, and the Web services description language description of the image cropping example as defined in Annex H.

I.5.2 Service profile

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE uridef [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema">
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema">
  <!ENTITY owl "http://www.w3.org/2002/07/owl">
  <!ENTITY profile "http://www.daml.org/services/owl-s/1.1/Profile.owl">
  <!ENTITY process "http://www.daml.org/services/owl-s/1.1/Process.owl">
  <!ENTITY hierarchy "urn:mpeg:mpeg21:2003:01-RDD-NS">
  <!ENTITY crop_process
    "http://www.itec.uni-klu.ac.at/~klaus/sws/stdcrop/CropProcess.owl">
  <!ENTITY DEFAULT
    "http://www.itec.uni-klu.ac.at/~klaus/sws/stdcrop/CropProfile.owl">
]>
<rdf:RDF
  xmlns:rdf="&rdf;#" xmlns:rdfs="&rdfs;#" xmlns:owl="&owl;#"
  xmlns:process="&process;#" xmlns:profile="&profile;#"
  xmlns:hierarchy="&hierarchy;#" xmlns:crop_process="&crop_process;#"
  xmlns="&DEFAULT;#">

  <owl:Ontology rdf:about="">
    <owl:versionInfo>
          $Id: CropSemProfile.owl,v 1.2 2004/10/20 klaus Exp $
    </owl:versionInfo>
    <rdfs:comment>
          OWL-S profile for an image cropper
    </rdfs:comment>
    <owl:imports rdf:resource="&profile;"/>
    <owl:imports rdf:resource="&process;"/>
    <owl:imports rdf:resource="&crop_process;"/>
  </owl:Ontology>

  <!-- Reference to the RDD term for the image crop conversion -->
  <hierarchy:CropRectangularBitmapImage rdf:ID="CropImage">
    <!-- reference to the process model specification -->
    <profile:has_process rdf:resource="&crop_process;#CropImage"/>
    <profile:serviceName>
   CropImage
    </profile:serviceName>
    <profile:textDescription>
   This conversion operation...
    </profile:textDescription>
    <profile:contactInformation>
   <!-- contact information described by, e.g., MPEG-7 -->
    </profile:contactInformation>
    <profile:hasInput rdf:resource="&crop_process;#roi-imageIn"/>
    <profile:hasInput rdf:resource="&crop_process;#roi-width"/>
    <profile:hasInput rdf:resource="&crop_process;#roi-height"/>
    <profile:hasInput rdf:resource="&crop_process;#roi-x"/>
    <profile:hasInput rdf:resource="&crop_process;#roi-y"/>
    <profile:hasOutput rdf:resource="&crop_process;#imageOut"/>
    <profile:hasEffect rdf:resource="&crop_process;#eff_resolution"/>
  </hierarchy:CropRectangularBitmapImage>

</rdf:RDF>
```

I.5.3 Service model

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE uridef [
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema">
  <!ENTITY owl "http://www.w3.org/2002/07/owl">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema">
  <!ENTITY process "http://www.daml.org/services/owl-s/1.1/Process.owl">
  <!ENTITY swrlx "http://www.csl.sri.com/users/denker/owl/swrl.owl">
  <!ENTITY owlx "http://www.w3.org/2003/05/owl-xml">
  <!ENTITY expr
    "http://www.daml.org/services/owl-s/1.1/generic/Expression.owl">
  <!ENTITY DEFAULT
    "http://www.itec.uni-klu.ac.at/~klaus/sws/stdcrop/CropProcess.owl">
]>
<rdf:RDF
  xmlns:rdf="&rdf;#" xmlns:rdfs="&rdfs;#" xmlns:owl="&owl;#"
  xmlns:xsd="&xsd;" xmlns:process="&process;#" xmlns:swrlx="&swrlx;#"
  xmlns:owlx="&owlx;#" xmlns:expr="&expr;#">

  <owl:Ontology rdf:about="">
    <owl:versionInfo>
   $Id: Crop Image V 1.2
    </owl:versionInfo>
    <rdfs:comment>
   Semantic description of a crop operation
    </rdfs:comment>
    <owl:imports rdf:resource="&rdf;"/>
    <owl:imports rdf:resource="&rdfs;"/>
    <owl:imports rdf:resource="&owl;"/>
    <owl:imports rdf:resource="&process;"/>
  </owl:Ontology>

  <!--Process-->
  <process:AtomicProcess rdf:ID="CropImage">
    <rdfs:label> The CropImage process </rdfs:label>

    <!-- The parameters of the process -->
    <process:hasInput>
      <process:Input rdf:ID="imageIn">
        <process:parameterType rdf:datatype="&xsd;#anyURI">
          <!-- JPEG image -->
          urn:mpeg:mpeg7:cs:FileFormatCS:2001:1
        </process:parameterType>
      </process:Input>
    </process:hasInput>

    <process:hasInput>
      <process:Input rdf:ID="roi-width">
        <process:parameterType rdf:datatype="&xsd;#anyURI">
          <!-- newly introduced VisualCS - region of interest:width -->
          urn:mpeg7:cs:VisualCS:2002:3.1
        </process:parameterType>
      </process:Input>
    </process:hasInput>

    <process:hasInput>
      <process:Input rdf:ID="roi-height">
        <process:parameterType rdf:datatype="&xsd;#anyURI">
          <!-- newly introduced VisualCS - region of interest:height -->
```