INTERNATIONAL STANDARD

ISO/IEC 23000-18

First edition
2018-05

Information technology — Multimedia application formats (MPEG-A) —

Part 18: Media linking application format

Technologies de l'information — Formats des applications multimédia (MPEG-A) —

Partie 18: Format des applications de liaison de médias

Reference number
ISO/IEC 23000-18:2018(E)

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

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

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

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

A list of all parts in the ISO/IEC 23000 series can be found on the ISO website.

Introduction

ISO/IEC 23000 (also known as “MPEG-A”) is an MPEG standard that supports a fast track to standardization by selecting readily tested and verified technologies taken from the MPEG corpus of standards and combining them to form a Multimedia Application Format (MAF). If a needed piece of technology is not available from the said corpus, then additional technologies originating from other organizations may also included by reference in order to facilitate the envisioned MAF.

The development of the MAF called “Media Linking Application Format” (MLAF) has been prompted by existing many examples of services where media transmitted for consumption on a primary device give hints to users to consume related media on a secondary or companion device. To facilitate interoperability of such services, it is beneficial to define a data structure (a “format”) that codifies the relationship between the two information sources.

The core of MLAF structure is the representation of the structure of bridgets. A bridget is a pivotal information element that links Source Content and Destination Content. A bridget is associated to its Presentation Information. Clause 7 specifies the representation of bridgets. Clause 8 specifies the presentation of bridgets.

The XML Schema of MLAF is defined in Annex A. Technology for bridget Presentation based on XMT is included in Annex B. Bridget presentation in BIFS is defined in Annex C. Annex D reports some examples.

Information technology — Multimedia application formats (MPEG-A) —

Part 18: Media linking application format

1 Scope

This document specifies a data structure called "bridget". A bridget is a link between a "source" content and a "destination" content. The bridget contains information on the source content and on the destination content, on the link between the two and on how the information contained in the bridget is to be presented to users consuming the source content in order to enable them to make considerate decisions about whether to consume the destination content.

2 Normative references

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

ISO/IEC 14496-11:2015, *Information technology — Coding of audio-visual objects — Part 11: Scene description and application engine*

3 Terms and definitions

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

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

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

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

3.1 MPEG-4 terminology

3.1.1
audiovisual scene
AV scene
set of audiovisual objects together with scene description information that defines their spatial and temporal attributes including behaviours resulting from object and user interactions

3.1.2
binary format for scene
BIFS
coded representation of a parametric scene description format

3.2 MPEG-7 terminology

3.2.1
segment
spatial or temporal unit of multimedia, for example, a temporal segment of video, or a segment of an image

3.2.2
audiovisual segment
element of MPEG-7 multimedia description schemes structure description tools describing a time-continuous arbitrary part (segment) of an audiovisual media

Note 1 to entry: See ISO/IEC 15938-5.

3.2.3
region
spatial unit of multimedia, for example, a 2D spatial region of an image, or a *moving region* (3.2.4) of video

3.2.4
moving region
region (3.2.3) changing its shape during a video timeline

3.3 MPEG-21 terminology

3.3.1
digital item identification
DII
standard used by MPEG-21 for identification of digital item and their *components* (3.3.5)

Note 1 to entry: See ISO/IEC 21000-3.

3.3.2
resource
individually identifiable asset such as a video or audio clip, an image or a text

Note 1 to entry: All resources shall be locatable via an unambiguous address.

3.3.3
container
structure that allows *items* (3.3.4) and/or containers to be grouped

Note 1 to entry: A container itself is not an item; containers are groupings of items and/or containers. Descriptors allow for the "labelling" of containers with information appropriate for the purpose of the grouping.

3.3.4
item
grouping of sub-items and/or *components* (3.3.5) that are bound to relevant *descriptors* (3.3.6)

Note 1 to entry: These descriptors contain information about the item. Items may contain choices, which allow them to be customized or configured. Items may be conditional (on predicates asserted by selections defined in the choices). An item that contains no sub-items can be considered a whole. An item that does contain sub-items can be considered a compilation. Items may also contain annotations to their sub-parts.

3.3.5
component
binding of a *resource* (3.3.2) to a set of *descriptors* (3.3.6)

Note 1 to entry: These descriptors are information concerning all or part of the specific resource instance. A component itself is not an item; components are building blocks of items. Components may be conditional (see ISO/IEC 21000-3 for details).

3.3.6
descriptor
associates information with the enclosing entity

Note 1 to entry: This information may be a component (such as a thumbnail of an image, or a text component) or a textual statement. Descriptors may be conditional (see ISO/IEC 21000-3 for details).

3.3.7
statement
literal textual value that contains information, but not an asset

EXAMPLE Examples of likely statements include descriptive, control, revision tracking or identifying information (such as an identifier as described in ISO/IEC 21000-3).

3.3.8
annotation
describes a set of information about another identified entity of the model without altering or adding to that entity

Note 1 to entry: The information can take the form of assertions, descriptors and anchors.

3.3.9
anchor
binds *descriptors* (3.3.6) to a *fragment,* which corresponds to a specific location or part of a *resource* (3.3.2)

Note 1 to entry: These descriptors are information concerning all or part of the fragment. Anchors may be conditional.

3.3.10
condition
describes the enclosing entity as being optional, and links it to the *selection*(s) that affect its inclusion

Note 1 to entry: Multiple predicates within a condition are combined as a conjunction (an AND relationship). Any predicate may be negated within a condition. Multiple conditions associated with a given entity are combined as a disjunction (an OR relationship) when determining whether to include the entity.

3.4 Terminology inherited from other standards — EBU Core terminology

3.4.1
core metadata
core descriptive information regarding a *resource* (3.3.2)

Note 1 to entry: Core descriptive information include, e.g., titles, contributors, genres, technical formats of a given resource.

3.4.2
metadata provider
entity (person or organization) providing metadata

3.5 MLAF terminology

3.5.1
bridget
link from a piece of *source content* (3.5.2) (e.g. a TV programme) to a piece of *destination content* (3.5.3) (e.g. an interactive media element such as a photograph, a multimedia presentation, a *segment* (3.2.1) or another TV programme, an application)

3.5.2
source content
content that is linked to *destination content* (3.5.3) through a *bridget* (3.5.1)

Note 1 to entry: The source content is the content from which the bridget originates.

3.5.3
destination content
content that is linked to the *source content* (3.5.2) through a *bridget* (3.5.1)

Note 1 to entry: The destination content is the content to which the bridget points.

3.5.4
spatiotemporal scope
information that describe the spatiotemporal *region* (3.2.3) of the linked *source content* (3.5.2) or of the linked *destination content* (3.5.3) that are involved in the link provided by a *bridget* (3.5.1)

3.5.5
global bridget
bridget (3.5.1) which is active for the whole duration of the *source content* (3.5.2)

3.5.6
spatiotemporal-aligned bridget
bridget (3.5.1) which is active for a determined *spatiotemporal scope* (3.5.4) of the *source content* (3.5.2)

4 Abbreviated terms

BIFS	BInary Format for Scenes
DID	MPEG-21, Digital Item Declaration
DIDL	MPEG-21, Digital Item Declaration Language
DII	MPEG-21, Digital Item Identification
MLAF	Media Linking Application Format
XMT	Extensible MPEG-4 Textual Format

5 MLAF model

5.1 General

Bridgets are links which exist because of some inherent semantic relationship between content items. As such, they can be products of an editorial decision, taken by someone as the result of the inspection (which can be manual or automatic) of content items, and can be products of a workflow which involves different roles taking care of finding, organizing and finally crafting the data that constitute them. The nature of a bridget is, however, quite different than traditional linearity of media content, and as such it induces a different, more "distributed" workflow. In fact, whether a piece of media content is a candidate source or destination for a bridget can be the result of an editorial decision taken at any moment and by quite different kind of users.

What is foreseeable is a sort of "layered" approach at producing bridgets in which actors with different roles define bridgets under different perspectives and possibly concurring at the same time. Authors of programmes will define bridget end points (i.e. sources and destination content items) following the criteria matching with the editorial intention, main distribution channel or target audience of the programme. At the same time, marketing and commercial operators (e.g. advertisement agents) will define such end points following their own mind-setting, which may be independent from the authorial perspective. Last, but definitely not least, final users can define their own ways for bridgets through social media interaction. All the above approaches can include not only the generation of the linking information but also of information related to how referenced content have to be presented graphically or should interact with the user.

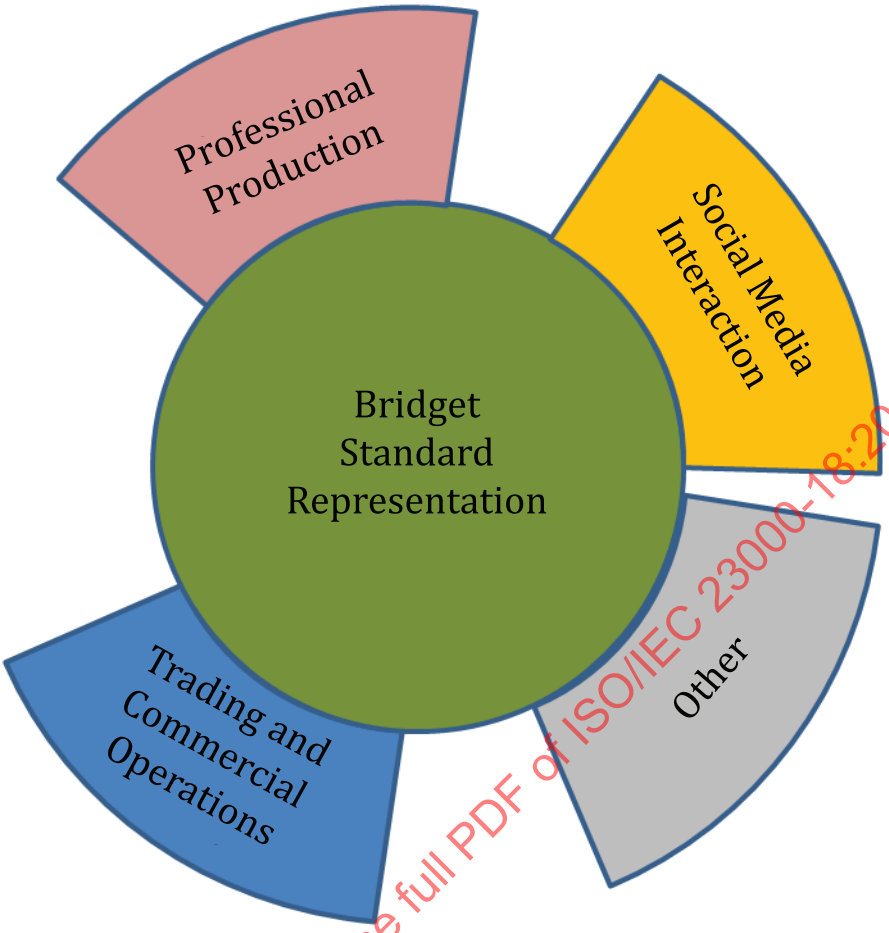

Figure 1— Bridget creation workflow

The result is that the way in which bridget information would be created is quite different from traditional linear approaches at media production, and more following a daisy chain paradigm (see Figure 1).

Therefore, a standard format for representing and exchanging bridget-related information fosters integration of all those systems having a role in generating bridget information in the different and heterogeneous aforementioned domains. This document specifies this standard format as Media Linking Application Format (MLAF).

5.2 Functional requirements

a) Source and Destination Content of a bridget can be

   1) a file or a stream, or

   2) a static medium (e.g. an image) or a dynamic medium (e.g. a video).

b) A bridget shall include a unique and persistent identifier.

c) A bridget may include authoring information (e.g. title, date, copyright, etc.).

d) A bridget shall include identification of its Source Content.

e) A bridget shall include information about the link between Source and Destination Content.

   1) A bridget shall include reference to

      i) a specific time instant of a dynamic medium representing the source or destination content, or

      ii) a specific time period of a dynamic medium representing the source or destination content, or

      iii) the entire duration of a dynamic medium representing the source or destination content.

   2) A bridget may include reference to

      i) a specific spatial region of a static medium representing a destination content, or

      ii) a specific spatiotemporal region (e.g. a 2D or 3D object) of a dynamic medium representing a destination content.

   3) A bridget may include metadata describing the context of the link between source and destination content (e.g. the semantics of the relation between source and destination).

f) A bridget may include metadata describing the Source Content:

   1) title, date, copyright, etc.;

   2) content type, media duration, MIME type, file size, etc.

g) A bridget may include textual descriptions of the Source Content.

h) A bridget shall include identification of Destination Content.

i) A bridget may include metadata describing the Destination Content:

   1) title, date, copyright, etc.;

   2) content type, media duration, MIME type, file size, etc.

j) A bridget may include textual descriptions of the Destination Content.

k) A bridget may include information on how a bridget should be presented.

l) A bridget may include an icon or image to visually present the bridget to the user.

m) A bridget may include information on visual/acoustic presentation of the Destination Content.

n) A bridget may include consumption-related information, e.g.

   1) parental rating;

   2) accessibility;

   3) device requirements (e.g. stereo headphones);

   4) target users.

o) A bridget may include rights and payment information for destination content consumption.

p) A bridget may include information about the service providing the destination content.

6 MLAF specification

6.1 Imported schemas and MLAF namespace URI

6.1.1 Schema header elements

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:didl="urn:mpeg:mp
eg21:2002:02-DIDL-NS" xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS" xmlns:mpeg7="ur
n:mpeg:mpeg7:schema:2004" xmlns:ebucore="urn:ebu:metadata-schema:ebuCore_2014" xmlns:ud="u
rn:mpeg:mpeg21:UD:UD:2016" xmlns:rd="urn:mpeg:mpeg21:UD:RD:2016" xmlns="urn:mpeg:mpega:mla
f:schema:2016" xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:schema:2016" targetNamespace="urn:mpeg
:mpega:mlaf:schema:2016" elementFormDefault="qualified" attributeFormDefault="unqualified">

    <xs:import namespace="urn:mpeg:mpeg21:2002:02-DIDL-NS" schemaLocation="http://
standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/did/didl.xsd"/>

    <xs:import namespace="urn:mpeg:mpeg7:schema:2004" schemaLocation="http://standards.
iso.org/ittf/PubliclyAvailableStandards/MPEG-7_schema_files/mpeg7-v2.xsd"/>

    <xs:import namespace="urn:ebu:metadata-schema:ebuCore_2014" schemaLocation="EBU_
CORE_20140318.xsd"/>

    <xs:import namespace="urn:mpeg:mpeg4:mlaf:schema:2016" schemaLocation="BridgetXMT/
bridget_mlaf-a.xsd"/>

    <xs:import namespace="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS" schemaLocation="http://
standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/did/didmodel.xsd"/>

    <xs:import namespace="urn:mpeg:mpeg21:UD:RD:2016" schemaLocation="MPEG-21%20Part%20
22/4.MPEG_21_UD_RD.xsd"/>


<!—The MLAF XML Schema definitions -->


</xs:schema>
```

6.1.2 MLAF namespace URI

The URI of the MLAF namespace for bridget Representation is urn:mpeg:mpega:mlaf:schema:2016. The URI of the MLAF namespace for bridget presentation is urn:mpeg:mpeg4:mlaf:schema:2016.

7 Bridget Representation

7.1 General

In this clause, the technologies for bridget Representation part of MLAF are presented. Bridget information (i.e. the source content, the destination content, the link and related descriptions) are represented through a restriction of MPEG-21 DIDL elements.

7.2 MLAF Root element

7.2.1 General

The root element of a MLAF document.

7.2.2 Syntax

```
<xs:element name="DIDL">
   <xs:annotation>
      <xs:documentation>The root element of a MLAF document</xs:documentation>
   </xs:annotation>
   <xs:complexType>
      <xs:complexContent>
         <xs:restriction base="didl:DIDLType">
            <xs:sequence>
               <xs:element ref="didl:DIDLInfo" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="didl:Declarations" minOccurs="0"/>
               <xs:element ref="MLAFContainer"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
</xs:element>
```

7.2.3 Semantics

Name	*Definition*
didl:DIDLType	As defined in ISO/IEC 21000-2
didl:DIDLInfo	As defined in ISO/IEC 21000-2
MLAFContainer	A MLAF Container

7.2.4 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<didl:DIDL xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpega:mlaf:schema:2016 MLAF.
xsd" xmlns:mlaf="urn:mpeg:mpega:mlaf:schema:2016" xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:sc
hema:2016">
    ...
</didl:DIDL>
```

7.3 MLAF Container

7.3.1 General

A restriction of MPEG-21 DIDL Container representing the main container for MLAF.

7.3.2 Syntax

```
<xs:complexType name="MLAFContainerType">
       <xs:annotation>
          <xs:documentation>A restriction of MPEG-21 DIDL Container representing the
main container for MLAF</xs:documentation>
       </xs:annotation>
       <xs:complexContent>
          <xs:restriction base="didl:ContainerType">
             <xs:sequence>
                <xs:element ref="didmodel:Descriptor" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:sequence minOccurs="1" maxOccurs="unbounded">
                   <xs:element ref="SourceItem" minOccurs="1" maxOccurs="1"/>
                   <xs:element ref="Bridgets"/>
                </xs:sequence>
             </xs:sequence>
          </xs:restriction>
       </xs:complexContent>
    </xs:complexType>
<xs:element name="MLAFContainer" type="MLAFContainerType" substitutionGroup="didl:Cont
ainer"/>
```

7.3.3 Semantics

Name	*Definition*
didmodel:Descriptor	As defined in ISO/IEC 21000-2
bridgets	A list of bridgets associated to the Source content
SourceItem	The Source content of the bridgets.

7.3.4 Example

```
    <mlaf:MLAFContainer>
       <mlaf:SourceItem>…</mlaf:SourceItem>
       <mlaf:Bridgets>
       ...
       </mlaf:Bridgets>
</mlaf:MLAFContainer>
```

7.4 Bridgets

7.4.1 General

A restriction of MPEG-21 DIDL Item representing the list of bridgets associated to a Source content.

7.4.2 Syntax

```
<xs:complexType name="BridgetsType" mixed="false">
       <xs:annotation>
          <xs:documentation>A restriction of MPEG-21 DIDL Item representing the container
of the Bridgets</xs:documentation>
       </xs:annotation>
       <xs:complexContent>
          <xs:restriction base="didl:ItemType">
             <xs:sequence>
                <xs:element ref="MainLayoutDescriptor" minOccurs="1"/>
                <xs:choice minOccurs="1" maxOccurs="unbounded">
                   <xs:element ref="Bridget" maxOccurs="1"/>
                </xs:choice>
             </xs:sequence>
          </xs:restriction>
       </xs:complexContent>
   </xs:complexType>
   <xs:element name="Bridgets" type="BridgetsType" substitutionGroup="didl:Item"/>
```

7.4.3 Semantics

Name	*Definition*
MainLayoutDescriptor	The descriptor of the main presentation layout of the bridget.
Bridget	A bridget

7.4.4 Example

```
    <mlaf:Bridgets>
          <mlaf:MainLayoutDescriptor>
             ...
          </mlaf:MainLayoutDescriptor>
       <mlaf:Bridget>...</mlaf:Bridget>
       <mlaf:Bridget>...</mlaf:Bridget>
       ...
</mlaf:Bridgets>
```

7.5 Source Item

7.5.1 General

A restriction of MPEG-21 DIDL Item representing the source content.

7.5.2 Syntax

```
<xs:complexType name="SourceItemType" mixed="false">
       <xs:annotation>
          <xs:documentation>A restriction of MPEG-21 DIDL Item representing the source
content</xs:documentation>
       </xs:annotation>
       <xs:complexContent>
          <xs:restriction base="didl:ItemType">
             <xs:sequence>
                <xs:element ref="didmodel:Condition" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element ref="MediaDescriptor" minOccurs="0" maxOccurs="unbounded"/>
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                   <xs:element ref="didmodel:Component"/>
                </xs:choice>
             </xs:sequence>
          </xs:restriction>
       </xs:complexContent>
    </xs:complexType>
    <xs:element name="SourceItem" type="SourceItemType" substitutionGroup="didl:Item"/>
```

7.5.3 Semantics

Name	*Definition*
didmodel:Condition	As defined in ISO/IEC 21000-2
MediaDescriptor	A media descriptor describing the source content
didmodel:Component	As defined in ISO/IEC 21000-2

7.5.4 Example

```
<mlaf:SourceItem>
          <mlaf:TargetCondition xmlns:rd="urn:mpeg:mpeg21:UD:RD:2016" xmlns:ud="urn:mpeg:
mpeg21:UD:UD:2016">
             <mlaf:RecommendedBridgetUsageCondition>
                <rd:UserDescriptionFragment>
                   ...
                </rd:UserDescriptionFragment>
             </mlaf:RecommendedBridgetUsageCondition>
          </mlaf:TargetCondition>
          <mlaf:MPEG7MediaDescriptor xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
             <mlaf:MPEG7Statement mimeType="text/xml">
                <mlaf:MPEG7ContentEntity>
                   <mpeg7:MultimediaContent xsi:type="mpeg7:VideoType">
                      <mpeg7:Video>...</mpeg7:Video>
                   </mpeg7:MultimediaContent>
                </mlaf:MPEG7ContentEntity>
             </mlaf:MPEG7Statement>
          </mlaf:MPEG7MediaDescriptor>
          <didl:Component>
             <didl:Resource mimeType="video/mpeg"></didl:Resource>
          </didl:Component>
</mlaf:SourceItem>
```

7.6 Bridget

7.6.1 General

A restriction of MPEG-21 DIDL Item representing the bridget. A Bridget is the implementation of the concept of bridget.

7.6.2 Syntax

```
<xs:complexType name="BridgetType" mixed="false">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Item representing the Bridget</
xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:ItemType">
            <xs:sequence>

               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="BridgetDescriptor" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="didmodel:Choice" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice minOccurs="1" maxOccurs="unbounded">
                  <xs:element ref="DestinationItem"/>
               </xs:choice>
               <xs:element ref="SourceItemStructuralAnnotation" minOccurs="1"
maxOccurs="1"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="Bridget" type="BridgetType" substitutionGroup="didl:Item"/>
```

7.6.3 Semantics

Name	*Definition*
didmodel:Condition	As defined in ISO/IEC 21000-2
BridgetDescriptor	A Descriptor of the bridget
didmodel:Choice	As defined in ISO/IEC 21000-2
DestinationItem	The destination content of the bridget.
SourceItemStructuralAnnotation	This element specifies the part of the Source content of this bridget in which the bridget is active. The part can be a temporal segment or a moving region.

7.6.4 Example

```
<mlaf:Bridget>
            <mlaf:DestinationItem>
               ...
            </mlaf:DestinationItem>
            <mlaf:DestinationItem>
               ...
            </mlaf:DestinationItem>
            <mlaf:DestinationItem>
               ...
            </mlaf:DestinationItem>
            <mlaf:SourceItemStructuralAnnotation target="">
            ...
            </mlaf:SourceItemStructuralAnnotation>
</mlaf:Bridget>
```

7.7 Destination Item

7.7.1 General

A restriction of MPEG-21 DIDL Item representing the destination content of a bridget.

7.7.2 Syntax

```
<xs:complexType name="DestinationItemType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Item representing the
destination content</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:ItemType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="MediaDescriptor" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice minOccurs="0" maxOccurs="unbounded">
                  <xs:element ref="didmodel:Component"/>
               </xs:choice>
               <xs:element ref="DestinationItemStructuralAnnotation" minOccurs="1"
maxOccurs="1"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="DestinationItem" type="DestinationItemType"
substitutionGroup="didl:Item"/>
```

7.7.3 Semantics

Name	*Definition*
didmodel:Condition	As defined in ISO/IEC 21000-2
MediaDescriptor	A Media Descriptor describing the Destination content
didmodel:Component	As defined in ISO/IEC 21000-2
DestinationItemStructuralAnnotation	This element specifies which spatiotemporal portion of the Destination content is linked to the Source content thorugh the enclosing bridget.

7.7.4 Example

```
<mlaf:DestinationItem>
         <mlaf:TargetCondition xmlns:rd="urn:mpeg:mpeg21:UD:RD:2016" xmlns:ud="urn:mpeg:m
peg21:UD:UD:2016">
            <mlaf:RecommendedBridgetUsageCondition>
               <rd:UserDescriptionFragment>
                  ...
               </rd:UserDescriptionFragment>
            </mlaf:RecommendedBridgetUsageCondition>
         </mlaf:TargetCondition>
         <mlaf:MPEG7MediaDescriptor xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
            <mlaf:MPEG7Statement mimeType="text/xml">
               <mlaf:MPEG7ContentEntity>
                  <mpeg7:MultimediaContent xsi:type="mpeg7:VideoType">
                     <mpeg7:Video>...</mpeg7:Video>
                  </mpeg7:MultimediaContent>
               </mlaf:MPEG7ContentEntity>
            </mlaf:MPEG7Statement>
         </mlaf:MPEG7MediaDescriptor>
         <didl:Component>
            <didl:Resource mimeType="video/mpeg"></didl:Resource>
         </didl:Component>
</mlaf:DestinationItem>
```

7.8 Media Segments

7.8.1 General

This complex type defines generic segments of media.

7.8.2 Syntax

```
<xs:complexType name="MediaSegmentType">
      <xs:annotation>
         <xs:documentation>This complex type defines generic segments of media</
xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="SpatioTemporalScope" maxOccurs="unbounded">
            <xs:annotation>
               <xs:documentation>This element carries information about a generic
spatiotemporal part of a piece of media. This can be expressed either as a Moving Region
or as an audiovisual segment</xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:choice>
                  <xs:element name="MovingRegion" type="mpeg7:MovingRegionSpatioTemporalD
ecompositionType">
                     <xs:annotation>
                        <xs:documentation>A moving (or still) region</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="AVSegment" type="mpeg7:AudioVisualSegmentType">
                     <xs:annotation>
                        <xs:documentation>An audiovisual temporal segment</
xs:documentation>
                     </xs:annotation>
                  </xs:element>
               </xs:choice>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
   <xs:element name="MediaSegments" type="MediaSegmentType"/>
```

7.8.3 Semantics

Name	*Definition*
SpatioTemporalScope	This element carries information about a generic spatiotemporal part of a piece of media. This can be expressed either as an MPEG-7 Moving Region or as an Audiovisual segment.
MovingRegion	An element of type mpeg7: MovingRegionSpatioTemporalDecompositionType specifying a moving region.
AVSegment	An audiovisual temporal segment of type mpeg7: AudioVisualSegmentType specifying a temporal segment.

7.8.4 Example

```
<mlaf:MediaSegments xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
      <mlaf:SpatioTemporalScope>
         <mlaf:MovingRegion>
            <mpeg7:MovingRegion>...</mpeg7:MovingRegion>
         </mlaf:MovingRegion>
      </mlaf:SpatioTemporalScope>
</mlaf:MediaSegments>
```

7.9 Structural Annotation

7.9.1 General

A generic complex type derived by restriction of MPEG-21 DIDL Annotation carrying spatiotemporal structural information of a media item representing a source or destination content.

7.9.2 Syntax

```
<xs:complexType name="StructuralAnnotationType">
      <xs:annotation>
         <xs:documentation>A generic complex type derived by restriction of MPEG-
21 DIDL Annotation carrying spatiotemporal structural information of a media item.</
xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:AnnotationType">
            <xs:sequence>
               <xs:element ref="didmodel:Assertion" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice minOccurs="0" maxOccurs="1">
                  <xs:element ref="StructuralAnchor" minOccurs="1"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
```

7.9.3 Semantics

Name	*Definition*
didmodel:Assertion	As defined in ISO/IEC 21000-22
StructuralAnchor	An Anchor carrying the structural information specifying a spatiotemporal region of a piece of media.

7.9.4 Example

See 7.12.4 and 7.13.3.

7.10 Structural Anchor

7.10.1 General

A restriction of MPEG-21 DIDL Anchor to carry specific information about media segments.

7.10.2 Syntax

```
<xs:complexType name="StructuralAnchorType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Anchor to carry specific
information about Source segments</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:AnchorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="SpatioTemporalInformation" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="StructuralAnchor" type="StructuralAnchorType" substitutionGroup="did
l:Anchor"/>
```

7.10.3 Semantics

Name	*Definition*
didmodel:Condition	As defined in ISO/IEC 21000-2
SpatioTemporalInformation	An element carrying information about the spatiotemporal information of a segment of media.

7.10.4 Example

```
<mlaf:StructuralAnchor>
      <mlaf:SpatioTemporalInformation>
         <mlaf:MediaSegments xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
            <mlaf:SpatioTemporalScope>
               <mlaf:MovingRegion>
                  <mpeg7:MovingRegion> ... </mpeg7:MovingRegion>
               </mlaf:MovingRegion>
            </mlaf:SpatioTemporalScope>
         </mlaf:MediaSegments>
      </mlaf:SpatioTemporalInformation>
</mlaf:StructuralAnchor>
```

7.11 Spatiotemporal Information

7.11.1 General

A container of media segments.

7.11.2 Syntax

```
<xs:complexType name="SpatiotemporalInformationType">
      <xs:annotation>
         <xs:documentation>A container of Media Segments</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:FragmentType">
            <xs:sequence>
               <xs:element ref="MediaSegments" minOccurs="0"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="SpatioTemporalInformation" type="SpatiotemporalInformationType" subs
titutionGroup="didl:Fragment"/>
```

7.11.3 Semantics

Name	*Definition*
MediaSegments	A list of media segments

7.11.4 Example

```
<mlaf:SpatioTemporalInformation>
   <mlaf:MediaSegments xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
      <mlaf:SpatioTemporalScope>
         <mlaf:MovingRegion>
         <mpeg7:MovingRegion> ... </mpeg7:MovingRegion>
         </MovingRegion>
      </mlaf:SpatioTemporalScope>
   </mlaf:MediaSegments>
</mlaf:SpatioTemporalInformation>
```

7.12 Source Item Structural Annotation

7.12.1 General

This descriptor specifies the spatiotemporal scope of a bridget.

7.12.2 Syntax

```
   <xs:element name="SourceItemStructuralAnnotation" type="StructuralAnnotationType" subs
titutionGroup="didl:Annotation"/>
```

7.12.3 Semantics

Name	*Definition*
SourceItemStructuralAnnotation	Element carrying information about in which spatiotemporal part of the Source content item the enclosing bridget is active (spatiotemporal scope).

7.12.4 Example

```
<mlaf:SourceItemStructuralAnnotation target="http://www.altova.com/">
   <mlaf:StructuralAnchor>
      <mlaf:SpatioTemporalInformation>
         <mlaf:MediaSegments xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
            <mlaf:SpatioTemporalScope>
               <mlaf:MovingRegion>
                  <mpeg7:MovingRegion> ... </mpeg7:MovingRegion>
               </mlaf:MovingRegion>
            </mlaf:SpatioTemporalScope>
         </mlaf:MediaSegments>
      </mlaf:SpatioTemporalInformation>
   </mlaf:StructuralAnchor>
</mlaf:SourceItemStructuralAnnotation>
```

7.13 Destination Item Structural Annotation

7.13.1 Syntax

```
<xs:element name="DestinationItemStructuralAnnotation" type="StructuralAnnotationType" su
bstitutionGroup="didl:Annotation"/>
```

7.13.2 Semantics

Name	*Definition*
DestinationItemStructuralAnnotation	Element carrying information about which spatiotemporal part of the Destination content item is linked through the enclosing bridget.

7.13.3 Example

```
<mlaf:DestinationItemStructuralAnnotation target="">
   <mlaf:StructuralAnchor>
      <mlaf:SpatioTemporalInformation>
         <mlaf:MediaSegments xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
            <mlaf:SpatioTemporalScope>
               <mlaf:AVSegment>
                  <mpeg7:TemporalDecomposition>
                     <mpeg7:AudioVisualSegment>...</mpeg7:AudioVisualSegment>
                  </mpeg7:TemporalDecomposition>
               </mlaf:AVSegment>
            </mlaf:SpatioTemporalScope>
         </mlaf:MediaSegments>
      </mlaf:SpatioTemporalInformation>
   </mlaf:StructuralAnchor>
</mlaf:DestinationItemStructuralAnnotation>
```

7.14 Media Descriptor

7.14.1 General

A restriction of MPEG-21 DIDL Descriptor representing a generic media descriptor (i.e. a descriptor of Source or Destination content).

7.14.2 Syntax

```
<xs:complexType name="MediaDescriptorType" abstract="true">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Descriptor representing a
generic media descriptor (i.e., a descriptor of Source  or Deestination content)</
xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:DescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="didmodel:Descriptor" minOccurs="0"
maxOccurs="unbounded"/>
               <xs:choice>
                  <xs:element ref="didmodel:Component"/>
                  <xs:element ref="didmodel:Statement"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MediaDescriptor" type="MediaDescriptorType" substitutionGroup="didl:
Descriptor"/>
```

7.14.3 Semantics

Name	*Definition*
didmodel:Condition	As defined in ISO/IEC 21000-2
didmodel:Descriptor	As defined in ISO/IEC 21000-2
didmodel:Component	As defined in ISO/IEC 21000-2
didmodel:Statement	As defined in ISO/IEC 21000-2

7.14.4 Example

This is an abstract compelx type. Refer to 7.16.4 and 7.19.4 for concrete examples.

7.15 Bridget Descriptor

7.15.1 General

A restriction of MPEG-21 DIDL Descriptor representing a generic bridget descriptor.

7.15.2 Syntax

```
<xs:complexType name="BridgetDescriptorType" abstract="true">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Descriptor representing a
generic Bridget descriptor</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:DescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="didmodel:Descriptor" minOccurs="0"
maxOccurs="unbounded"/>
               <xs:choice>
                  <xs:element ref="didmodel:Component"/>
                  <xs:element ref="didmodel:Statement"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetDescriptor" type="BridgetDescriptorType" substitutionGroup="d
idl:Descriptor"/>
```

7.15.3 Semantics

Name	*Definition*
didmodel:Condition	As defined in ISO/IEC 21000-2
didmodel:Descriptor	As defined in ISO/IEC 21000-2
didmodel:Component	As defined in ISO/IEC 21000-2
didmodel:Statement	As defined in ISO/IEC 21000-2

7.15.4 Example

This is an abstract compelx type. Refer to 7.17.4 for a concrete example.

7.16 EBU Core Media Descriptor

7.16.1 General

A media descriptor structured in EBU Core[16].

7.16.2 Syntax

```
<xs:complexType name="EBUCoreMediaDescriptorType">
      <xs:annotation>
         <xs:documentation>A media descriptor structured in EBU Core</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="MediaDescriptorType">
            <xs:sequence>
               <xs:element ref="EBUCoreStatement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
```

7.16.3 Semantics

Name	*Definition*
EBUCoreStatement	An EBU Core Statement

7.16.4 Example

```
<mlaf:MediaDescriptor xsi:type="EBUCoreMediaDescriptorType" xmlns:ebucore="
urn:ebu:metadata-schema:ebuCore_2014">
   <mlaf:EBUCoreStatement mimeType="text/xml">
      <ebucore:ebuCoreMain>
         <ebucore:coreMetadata>...</ebucore:coreMetadata>
      </ebucore:ebuCoreMain>
   </mlaf:EBUCoreStatement>
</mlaf:MediaDescriptor>
```

7.17 EBU Core bridget Descriptor

7.17.1 General

A bridget descriptor structured id EBU Core[16].

7.17.2 Syntax

```
<xs:complexType name="EBUCoreBridgetDescriptorType">
      <xs:annotation>
         <xs:documentation>A Bridget descriptor structured id EBU Core</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="BridgetDescriptorType">
            <xs:sequence>
               <xs:element ref="EBUCoreStatement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
```

7.17.3 Semantics

Name	*Definition*
EBUCoreStatement	An EBU Core Statement

7.17.4 Example

```
<mlaf:BridgetDescriptor xsi:type="EBUCoreBridgetDescriptorType">
      <mlaf:EBUCoreStatement mimeType="text/xml">
         <ebucore:ebuCoreMain>
            <ebucore:coreMetadata>...</ebucore:coreMetadata>
         </ebucore:ebuCoreMain>
      </mlaf:EBUCoreStatement>
   </mlaf:BridgetDescriptor>
```

7.18 EBU Core Statement

7.18.1 General

A restriction of MPEG-21 DIDL Statement carrying EBU Core descriptors.

7.18.2 Syntax

```
<xs:complexType name="EBUCoreStatementType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Statement carrying EBU Core
descriptors</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element ref="ebucore:ebuCoreMain" minOccurs="0"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="EBUCoreStatement" type="EBUCoreStatementType" substitutionGroup="did
l:Statement"/>
```

7.18.3 Semantics

Name	*Definition*
ebucore:ebuCoreMain	The root element of EBU Core[16].

7.18.4 Example

```
<mlaf:EBUCoreStatement mimeType="text/xml">
         <ebucore:ebuCoreMain>
            <ebucore:coreMetadata>...</ebucore:coreMetadata>
         </ebucore:ebuCoreMain>
      </mlaf:EBUCoreStatement>
```

7.19 MPEG-7 media descriptor

7.19.1 General

A media descriptor structured in ISO/IEC 15938-5[9].

7.19.2 Syntax

```
<xs:complexType name="MPEG7MediaDescriptorType">
      <xs:annotation>
         <xs:documentation> </xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="MediaDescriptorType">
            <xs:sequence>
               <xs:element ref="MPEG7Statement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MPEG7MediaDescriptor" type="MPEG7MediaDescriptorType" substitutionGr
oup="MediaDescriptor"/>
```

7.19.3 Semantics

Name	*Definition*
MPEG7Statement	An ISO/IEC 15938-5 statement

7.19.4 Example

```
<mlaf:MediaDescriptor xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004" xsi:type="MPEG7MediaDescri
ptorType">
   <mlaf:MPEG7Statement mimeType="text/xml">
      <mlaf:MPEG7ContentEntity>
         <mpeg7:MultimediaContent>...</mpeg7:MultimediaContent>
      </mlaf:MPEG7ContentEntity>
   </mlaf:MPEG7Statement>
</mlaf:MediaDescriptor>
```

7.20 MPEG-7 Statement

7.20.1 General

A restriction of MPEG-21 DIDL Statement carrying ISO/IEC 15938-5 descriptors.

7.20.2 Syntax

```
<xs:complexType name="MPEG7StatementType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Statement carrying MPEG-7 Part 5
descriptors</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element name="MPEG7ContentEntity" type="mpeg7:ContentEntityType"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MPEG7Statement" type="MPEG7StatementType" substitutionGroup="didl:St
atement"/>
```

7.20.3 Semantics

Name	*Definition*
MPEG7ContentEntity	A description of a Content Entity of type mpeg7:ContentEntityType

7.20.4 Example

```
<mlaf:MPEG7Statement mimeType="text/xml">
      <mlaf:MPEG7ContentEntity>
         <mpeg7:MultimediaContent>...</mpeg7:MultimediaContent>
      </mlaf:MPEG7ContentEntity>
</mlaf:MPEG7Statement>
```

7.21 Bridget Multimedia Representation Descriptor

7.21.1 General

A restriction of MPEG-21 DIDL Descriptor representing a visual or aural representation of a bridget.

7.21.2 Syntax

```
<xs:complexType name="BridgetMultimediaRepresentationDescriptorType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Descriptor representing a visual
or aural representation of a Bridget</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="BridgetDescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="BridgetMultimediaRepresentationStatement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetMultimediaRepresentationDescriptor" type="BridgetMultimediaRe
presentationDescriptorType" substitutionGroup="BridgetDescriptor"/>
```

7.21.3 Semantics

Name	*Definition*
didmodel:Condition	As defined in ISO/IEC 21000-2
BridgetMultimediaRepresentation-Statement	A Bridget Multimedia Representation Statement

7.21.4 Example

```
<mlaf:BridgetMultimediaRepresentationDescriptor>
   <mlaf:BridgetMultimediaRepresentationStatement mimeType="">
      <mlaf:MultimediaRepresentation>
         <mpeg7:MediaUri>http://my.mediauri.org/010101</mpeg7:MediaUri>
      </mlaf:MultimediaRepresentation>
   </mlaf:BridgetMultimediaRepresentationStatement>
</mlaf:BridgetMultimediaRepresentationDescriptor>
```

7.22 Bridget Multimedia Representation Statement

7.22.1 General

A restriction of MPEG-21 DIDL Statement carrying visual or aural representation of a bridget.

7.22.2 Syntax

```
<xs:complexType name="BridgetMultimediaRepresentationStatementType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Statement carrying visual or
aural representation of a Bridget</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element name="MultimediaRepresentation"
type="mpeg7:MediaLocatorType"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetMultimediaRepresentationStatement" type="BridgetMultimediaRep
resentationStatementType" substitutionGroup="didl:Statement"/>
```

7.22.3 Semantics

Name	*Definition*
MultimediaRepresentation	A multimedia representation of a bridget, structured as mpeg7:MediaLocatorType

7.22.4 Example

```
<mlaf:BridgetMultimediaRepresentationStatement mimeType="text/xml">
      <mlaf:MultimediaRepresentation>
         <mpeg7:MediaUri>http://my.mediauri.org/010101</mpeg7:MediaUri>
      </mlaf:MultimediaRepresentation>
</mlaf:BridgetMultimediaRepresentationStatement>
```

7.23 Target Condition

7.23.1 General

An extension of MPEG-21 DIDL Condition carrying information about target users of the bridget and the recommended context of usage. The information about target users and context of usage is represented using a structure of type rd:RecommendableResourceType from ISO/IEC 21000-22.

7.23.2 Syntax

```
<xs:element name="TargetCondition" substitutionGroup="didl:Condition">
      <xs:annotation>
         <xs:documentation>An extension of MPEG-21 DIDL Condition carrying information
about target users of the Bridget</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:complexContent>
            <xs:extension base="didl:ConditionType">
               <xs:sequence maxOccurs="unbounded">
                  <xs:element name="RecommendedBridgetUsageCondition" type="rd:Recommenda
bleResourceType"/>
               </xs:sequence>
            </xs:extension>
         </xs:complexContent>
      </xs:complexType>
   </xs:element>
```

7.23.3 Semantics

Name	*Definition*
RecommendedBridgetUsageCondition	An element of type rd:RecommendableResourceType from ISO/IEC 21000-22

7.23.4 Example

In this example, a bridget is recommended for users speaking Italian.

```
<mlaf:TargetCondition xmlns:rd="urn:mpeg:mpeg21:UD:RD:2016" xmlns:ud="urn:mpeg:mpeg21:UD:
UD:2016">
   <mlaf:RecommendedBridgetUsageCondition>
      <rd:UserDescriptionFragment>
         <ud:UserID>u100</ud:UserID>
         <ud:UserProfile xmlns:ud="urn:mpeg:mpeg21:UD:UD:2016" xsi:type="ud:PersonProfileTy
pe">
            <ud:Language>
               <ud:Name>it</ud:Name>
            </ud:Language>
         </ud:UserProfile>
      </rd:UserDescriptionFragment>
   </mlaf:RecommendedBridgetUsageCondition>
</mlaf:TargetCondition>
```

8 Bridget Presentation

8.1 General

This clause specifies the technology for the presentation of bridgets.The bridget Presentation is capable of representing 2D and 3D scenes that may contain multimedia content such as 2D and/or 3D graphic elements, images, videos and sounds. The multimedia content can be part of the scene (stored locally) or it can be referenced (through hyperlinks) from external repositories where compliant media files are available. Moreover, a scene can be static or animated and it can be connected to sensory information that can be either local or remote. The bridget Presentation supports external interaction allowing the user to communicate with the scene elements and therefore affecting the scene behaviour. The actions that can happen in a scene can be programmatically triggered by using time sensors, routes and scripting while other actions can be triggered by the user interaction through touch sensors.

Annex B presents the normative way of describing bridget Presentation information based on ISO/IEC 14496-11.

Annex C presents the MPEG-4 (BIFS) nodes that are used to express a bridget Presentation.

8.2 Layout Ref

8.2.1 General

Specifies a bridget layout which can be either the bridgeted media layout itself (scene graph) or a URI to a file where the layout is described.

8.2.2 Syntax

```
<xs:complexType name="LayoutRefType">
      <xs:annotation>
         <xs:documentation>This complex type defines the layout presentation of a Bridget
destination</xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="BridgetLayoutUrl">
            <xs:complexType>
               <xs:simpleContent>
                  <xs:extension base="xs:anyURI">
                     <xs:attribute name="mimeType" type="xs:string"/>
                  </xs:extension>
               </xs:simpleContent>
            </xs:complexType>
         </xs:element>
         <xs:element name="BridgetLayoutScene" type="BridgetLayoutSceneType"/>
      </xs:sequence>
```

8.2.3 </xs:complexType>Semantics

Name	*Definition*
BridgetLayoutUrl	An element of type xs:anyURI pointing to a file where a bridget presentation is described.
@mimeType	The MIME type of the file where the bridget presentation is described (e.g. text/html).
BridgetLayoutScene	An element of type BridgetLayoutSceneType where the description of a bridget presentation can be defined locally.

8.2.4 Example

```
<mlaf:BridgetLayout xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:schema:2016">
      <mlaf:BridgetLayoutScene>
         <mlaf-pres:Scene>
            <mlaf-pres:Group>...</mlaf-pres:Group>
         </mlaf-pres:Scene>
   </mlaf:BridgetLayoutScene>
</mlaf:BridgetLayout>
```

8.3 Main Layout Ref

8.3.1 General

Specifies an element that allows a bridget player to link a bridget layout representation to the main scene graph of a bridgeted programme as a subset of it.

8.3.2 Syntax

```
<xs:complexType name="MainLayoutRefType">
      <xs:annotation>
         <xs:documentation>This complex type links the bridget layout representation to
the main programme layout, where the bridget presentation takes place</xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="MainLayoutUrl" type="mlaf:Inline"/>
         <xs:element name="MainLayoutScene" type="mlaf:TransformType"/>
      </xs:sequence>
   </xs:complexType>
```

8.3.3 Semantics

Name	*Definition*
MainLayoutUrl	An element of type mlaf:Inline that can inline a bridget layout representation to the main scene graph of the bridgeted programme.
MainLayoutScene	An element of type mlaf:TransformType that allows to connect a bridget layout representation to the main scene graph of the bridgeted programme.

8.3.4 Example

```
<mlaf:MainLayout xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:schema:2016">
   <mlaf:BridgetLayoutUrl>http://my.layouts.org/34208</BridgetLayoutUrl>
   <mlaf:BridgetLayoutScene>
      <mlaf-pres:Scene>
         <mlaf-pres:Group>...</mlaf-pres:Group>
      </mlaf-pres:Scene>
   </mlaf:BridgetLayoutScene>
   <mlaf:MainLayoutUri>
      <mlaf-pres:IS>
         <mlaf-pres:connect nodeField="..." protoField="...">...</mlaf-pres:connect>
      </mlaf-pres:IS>
   </mlaf:MainLayoutUri>
</mlaf:MainLayout>
```

8.4 Bridget Layout Scene

8.4.1 General

Describes how a bridget layout should be represented in order for a bridget player to be able to interpret the information and present the bridget. The graphical elements used to describe a bridget layout are defined in the attached XML schema and they are a subset of the elements defined in ISO/IEC 14496-11.

8.4.2 Syntax

```
<xs:complexType name="BridgetLayoutSceneType">
      <xs:annotation>
         <xs:documentation>This complex type defines the XMT layouting information of a
Bridget</xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element ref="mlaf: Scene"/>
      </xs:sequence>
   </xs:complexType>
```

8.4.3 Semantics

Name	Definition
mlaf: Scene	An element (a structure of elements) derived from the type xmta:Scene defined in ISO/IEC 14496-11 by restricting the set of elements.

8.4.4 Example

```
<mlaf:BridgetLayoutScene xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:schema:2016">
   <mlaf-pres:Scene>
      <mlaf-pres:Group>…</mlaf-pres:Group>
   </mlaf-pres:Scene>
</mlaf:BridgetLayoutScene>
```

8.5 Bridget Layout Descriptor

8.5.1 General

A restriction of MPEG-21 DIDL Descriptor representing a bridget Layout information.

8.5.2 Syntax

```
<xs:complexType name="BridgetLayoutDescriptorType">
      <xs:complexContent>
         <xs:restriction base="BridgetDescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="BridgetLayoutStatement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetLayoutDescriptor" type="BridgetLayoutDescriptorType" substitu
tionGroup="BridgetDescriptor"/>
```

8.5.3 Semantics

Name	Definition
didmodel:Condition	As defined in ISO/IEC 21000-2
BridgetLayoutStatement	A bridget Layout Statement

8.5.4 Example

```
<mlaf:BridgetLayoutDescriptor xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:schema:2016">
   <mlaf:BridgetLayoutStatement mimeType="text/xml">
      <mlaf:BridgetLayout>
         <mlaf:BridgetLayoutScene>
            <mlaf-pres:Scene>
               <mlaf-pres:Group>...</mlaf-pres:Group>
            </mlaf-pres:Scene>
         </mlaf:BridgetLayoutScene>
      </mlaf:BridgetLayout>
   </mlaf:BridgetLayoutStatement>
</mlaf:BridgetLayoutDescriptor>
```

8.6 Bridget Layout Statement

8.6.1 General

A restriction of MPEG-21 DIDL Statement carrying bridget Layout descriptors.

8.6.2 Syntax

```
<xs:complexType name="BridgetLayoutStatementType">
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element name="BridgetLayout" type="LayoutRefType"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetLayoutStatement" type="BridgetLayoutStatementType" substitut
ionGroup="didl:Statement"/>
```

8.6.3 Semantics

Name	Definition
BridgetLayout	An element of type LayoutRefType

8.6.4 Example

```
<mlaf:BridgetLayoutStatement mimeType="text/xml" xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:sch
ema:2016">
      <mlaf:BridgetLayout>
         <mlaf:BridgetLayoutScene>
            <mlaf-pres:Scene>
               <mlaf-pres:Group>...</mlaf-pres:Group>
            </mlaf-pres:Scene>
         </mlaf:BridgetLayoutScene>
      </mlaf:BridgetLayout>
</mlaf:BridgetLayoutStatement>
```

8.7 Main Layout Descriptor

8.7.1 General

A restriction of MPEG-21 DIDL Descriptor representing a Main Bridget Layout information.

8.7.2 Syntax

```
<xs:complexType name="MainLayoutDescriptorType">
      <xs:complexContent>
         <xs:restriction base="didl:DescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice>
                  <xs:element ref="MainLayoutStatement"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MainLayoutDescriptor" type="MainLayoutDescriptorType" substitutionGr
oup="didl:Descriptor"/>
```

8.7.3 Semantics

Name	Definition
didmodel:Condition	As defined in ISO/IEC 21000-2
MainLayoutStatement	A Main Layout Statement

8.7.4 Example

```
<mlaf:MainLayoutDescriptor xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:schema:2016">
   <mlaf:MainLayoutStatement mimeType="text/xml">
      <mlaf:MainLayout>
         <mlaf:BridgetLayoutUrl>http://www.mylayouts.org/482</mlaf:BridgetLayoutUrl>
         <mlaf:BridgetLayoutScene>
            <mlaf-pres:Scene>
               <mlaf-pres:Group>...</mlaf-pres:Group>
            </mlaf-pres:Scene>
         </mlaf:BridgetLayoutScene>
         <mlaf:MainLayoutUri>
            <mlaf-pres:IS>
               <mlaf-pres:connect nodeField="…" protoField="…">…</mlaf-pres:connect>
            </mlaf-pres:IS>
         </mlaf:MainLayoutUri>
      </mlaf:MainLayout>
   </mlaf:MainLayoutStatement>
</mlaf:MainLayoutDescriptor>
```

8.8 Main Layout Statement

8.8.1 General

A restriction of MPEG-21 DIDL Statement carrying Main bridget Layout descriptors.

8.8.2 Syntax

```
<xs:complexType name="MainLayoutStatementType">
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element name="MainLayout">
                  <xs:complexType>
                     <xs:complexContent>
                        <xs:extension base="LayoutRefType">
                           <xs:choice>
                              <xs:element name="MainLayoutUri" type="mlaf:InlineType"/>
                              <xs:element name="MainLayoutScene"
type="mlaf:TransformType"/>
                           </xs:choice>
                        </xs:extension>
                     </xs:complexContent>
                  </xs:complexType>
               </xs:element>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MainLayoutStatement" type="MainLayoutStatementType" substitutionGrou
p="didl:Statement"/>
```


8.8.3 Semantics

Name	Definition
MainLayout	An element containing URI or inline definition of Main Layout
MainLayoutUri	An URI pointing to a Main Layout
MainLayoutScene	Inline description of Main Layout

8.8.4 Example

```
<mlaf:MainLayoutStatement mimeType="text/xml"  xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:schem
a:2016">
   <mlaf:MainLayout>

   <mlaf:BridgetLayoutUrl>http://www.mylayout.org/849</mlaf:BridgetLayoutUrl>
      <mlaf-pres:BridgetLayoutScene>
         <mlaf-pres:Scene>
            <mlaf-pres:Group>...</mlaf-pres:Group>
         </mlaf-pres:Scene>
      </mlaf:BridgetLayoutScene>
      <mlaf:MainLayoutUri>
         <mlaf-pres:IS>
            <mlaf-pres:connect nodeField="…" protoField="…">…</mlaf-pres:connect>
         </mlaf-pres:IS>
      </mlaf:MainLayoutUri>
   </mlaf:MainLayout>
</mlaf:MainLayoutStatement>
```

Annex A
(normative)

MLAF XML schema

This annex includes the main MLAF XML schema (**MLAF.xsd**). This schema imports all the referenced normative schemas from other standards and the schemas for bridget presentation referenced in Annex B. As for the schema imported from ISO/IEC 21000-22, the recommended procedure is to download the latest schemas from http://standards.iso.org/iso-iec/21000/-22/ed-1/en/amd/1 and to place the downloaded schemas in a local folder named "MPEG-21 Part 22".

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:didl="urn:mpeg:mp
eg21:2002:02-DIDL-NS" xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS" xmlns:mpeg7="u
rn:mpeg:mpeg7:schema:2004" xmlns:ebucore="urn:ebu:metadata-schema:ebuCore_2014" xmlns:ud=
"urn:mpeg:mpeg21:UD:UD:2016" xmlns:rd="urn:mpeg:mpeg21:UD:RD:2016" xmlns="urn:mpeg:mpega:
mlaf:schema:2016" xmlns:mlaf-pres="urn:mpeg:mpeg4:mlaf:schema:2016" targetNamespace="urn:
mpeg:mpega:mlaf:schema:2016" elementFormDefault="qualified" attributeFormDefault="unqualif
ied">
   <xs:import namespace="urn:mpeg:mpeg21:2002:02-DIDL-NS" schemaLocation="http://
standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/did/didl.xsd"/>
   <xs:import namespace="urn:mpeg:mpeg7:schema:2004" schemaLocation="http://standards.
iso.org/ittf/PubliclyAvailableStandards/MPEG-7_schema_files/mpeg7-v2.xsd"/>
   <xs:import namespace="urn:ebu:metadata-schema:ebuCore_2014" schemaLocation="EBU_
CORE_20140318.xsd"/>
   <xs:import namespace="urn:mpeg:mpeg4:mlaf:schema:2016" schemaLocation="BridgetXMT/
bridget_mlaf-a.xsd"/>
   <xs:import namespace="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS" schemaLocation="http://
standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/did/didmodel.xsd"/>
   <xs:import namespace="urn:mpeg:mpeg21:UD:RD:2016" schemaLocation="MPEG-21%20Part%20
22/4.MPEG_21_UD_RD.xsd"/>
   <xs:element name="DIDL">
      <xs:annotation>
         <xs:documentation>The root element of a MLAF document</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:complexContent>
            <xs:restriction base="didl:DIDLType">
               <xs:sequence>
                  <xs:element ref="didl:DIDLInfo" minOccurs="0" maxOccurs="unbounded"/>
                  <xs:element ref="didl:Declarations" minOccurs="0"/>
                  <xs:element ref="MLAFContainer"/>
               </xs:sequence>
            </xs:restriction>
         </xs:complexContent>
      </xs:complexType>
   </xs:element>
   <xs:complexType name="MLAFContainerType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Container representing the main
container for MLAF</xs:documentation>
```

```
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:ContainerType">
            <xs:sequence>
               <xs:element ref="didmodel:Descriptor" minOccurs="0" maxOccurs="unbounded"/>
               <xs:sequence minOccurs="1" maxOccurs="unbounded">
                  <xs:element ref="SourceItem" minOccurs="1" maxOccurs="1"/>
                  <xs:element ref="Bridgets"/>
               </xs:sequence>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MLAFContainer" type="MLAFContainerType" substitutionGroup="didl:Containe
r"/>
   <xs:complexType name="BridgetsType" mixed="false">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Item representing the container of
the Bridgets</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:ItemType">
            <xs:sequence>
               <xs:element ref="MainLayoutDescriptor" minOccurs="1"/>
               <xs:choice minOccurs="1" maxOccurs="unbounded">
                  <xs:element ref="Bridget" maxOccurs="1"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="Bridgets" type="BridgetsType" substitutionGroup="didl:Item"/>
   <xs:complexType name="SourceItemType" mixed="false">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Item representing the source
content</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:ItemType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="MediaDescriptor" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice minOccurs="0" maxOccurs="unbounded">
                  <xs:element ref="didmodel:Component"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="SourceItem" type="SourceItemType" substitutionGroup="didl:Item"/>
   <xs:complexType name="BridgetType" mixed="false">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Item representing the Bridget</
xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:ItemType">
            <xs:sequence>
               <!--<xs:element ref="didmodel:Condition" minOccurs="0"
maxOccurs="unbounded"/>-->
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="BridgetDescriptor" minOccurs="0" maxOccurs="unbounded"/>
```

```
               <xs:element ref="didmodel:Choice" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice minOccurs="1" maxOccurs="unbounded">
                  <xs:element ref="DestinationItem"/>
               </xs:choice>
               <xs:element ref="SourceItemStructuralAnnotation" minOccurs="1"
maxOccurs="1"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="Bridget" type="BridgetType" substitutionGroup="didl:Item"/>
   <xs:complexType name="DestinationItemType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Item representing the
destination content</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:ItemType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="MediaDescriptor" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice minOccurs="0" maxOccurs="unbounded">
                  <xs:element ref="didmodel:Component"/>
               </xs:choice>
               <xs:choice>
                  <xs:element ref="DestinationItemStructuralAnnotation" maxOccurs="1"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="DestinationItem" type="DestinationItemType"
substitutionGroup="didl:Item"/>
   <xs:complexType name="MediaSegmentType">
      <xs:annotation>
         <xs:documentation>This complex type defines generic segments of media</
xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="SpatioTemporalScope" maxOccurs="unbounded">
            <xs:annotation>
               <xs:documentation>This element carries information about a generic
spatiotemporal part of a piece of media. This can be expressed either as a Moving Region
or as an audiovisual segment</xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:choice>
                  <xs:element name="MovingRegion" type="mpeg7:MovingRegionSpatioTemporalD
ecompositionType">
                     <xs:annotation>
                        <xs:documentation>A moving (or still) region</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="AVSegment" type="mpeg7:AudioVisualSegmentType">
                     <xs:annotation>
                        <xs:documentation>An audiovisual temporal segment</
xs:documentation>
                     </xs:annotation>
```

```
                </xs:element>
             </xs:choice>
          </xs:complexType>
       </xs:element>
    </xs:sequence>
 </xs:complexType>
 <xs:element name="MediaSegments" type="MediaSegmentType"/>
 <xs:complexType name="StructuralAnnotationType">
    <xs:annotation>
       <xs:documentation>A generic complex type derived by restriction of MPEG-
21 DIDL Annotation carrying spatiotemporal structural information of a media item.</
xs:documentation>
    </xs:annotation>
    <xs:complexContent>
       <xs:restriction base="didl:AnnotationType">
          <xs:sequence>
             <xs:element ref="didmodel:Assertion" minOccurs="0" maxOccurs="unbounded"/>
             <xs:choice minOccurs="0" maxOccurs="1">
                <xs:element ref="StructuralAnchor" minOccurs="1"/>
             </xs:choice>
          </xs:sequence>
       </xs:restriction>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="SourceItemStructuralAnnotation" type="StructuralAnnotationType" subs
titutionGroup="didl:Annotation"/>
 <xs:element name="DestinationItemStructuralAnnotation" type="StructuralAnnotationType"
substitutionGroup="didl:Annotation"/>
 <xs:complexType name="StructuralAnchorType">
    <xs:annotation>
       <xs:documentation>A restriction of MPEG-21 DIDL Anchor to carry specific
information about Source segments</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
       <xs:restriction base="didl:AnchorType">
          <xs:sequence>
             <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element ref="SpatioTemporalInformation" minOccurs="1" maxOccurs="1"/>
          </xs:sequence>
       </xs:restriction>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="StructuralAnchor" type="StructuralAnchorType" substitutionGroup="did
l:Anchor"/>
 <xs:complexType name="SpatiotemporalInformationType">
    <xs:annotation>
       <xs:documentation>A container of Media Segments</xs:documentation>
    </xs:annotation>
    <xs:complexContent>
       <xs:restriction base="didl:FragmentType">
          <xs:sequence>
             <xs:element ref="MediaSegments" minOccurs="0"/>
          </xs:sequence>
       </xs:restriction>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="SpatioTemporalInformation" type="SpatiotemporalInformationType" subs
titutionGroup="didl:Fragment"/>
 <xs:complexType name="MediaDescriptorType" abstract="true">
    <xs:annotation>
```

```
         <xs:documentation>A restriction of MPEG-21 DIDL Descriptor representing a generic
media descriptor (i.e., a descriptor of Source  or Deestination content)</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:DescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="didmodel:Descriptor" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice>
                  <xs:element ref="didmodel:Component"/>
                  <xs:element ref="didmodel:Statement"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MediaDescriptor" type="MediaDescriptorType" substitutionGroup="didl:De
scriptor"/>
   <xs:complexType name="BridgetDescriptorType" abstract="true">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Descriptor representing a generic
Bridget descriptor</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:DescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="didmodel:Descriptor" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice>
                  <xs:element ref="didmodel:Component"/>
                  <xs:element ref="didmodel:Statement"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetDescriptor" type="BridgetDescriptorType" substitutionGroup="did
l:Descriptor"/>
   <!--Descriptors derived from EBU Core 1.5-->
   <xs:complexType name="EBUCoreMediaDescriptorType">
      <xs:annotation>
         <xs:documentation>A media descriptor structured in EBU Core</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="MediaDescriptorType">
            <xs:sequence>
               <xs:element ref="EBUCoreStatement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="EBUCoreBridgetDescriptorType">
      <xs:annotation>
         <xs:documentation>A Bridget descriptor structured id EBU Core</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="BridgetDescriptorType">
            <xs:sequence>
               <xs:element ref="EBUCoreStatement"/>
            </xs:sequence>
```

```
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="EBUCoreStatementType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Statement carrying EBU Core
descriptors</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element ref="ebucore:ebuCoreMain" minOccurs="0"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="EBUCoreStatement" type="EBUCoreStatementType" substitutionGroup="didl:Stat
ement"/>
   <!--Media Descriptors derived from MPEG-7 AVDP-->
   <xs:complexType name="MPEG7MediaDescriptorType">
      <xs:annotation>
         <xs:documentation>A media descriptor structured in MPEG-7 Part 5</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="MediaDescriptorType">
            <xs:sequence>
               <xs:element ref="MPEG7Statement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MPEG7MediaDescriptor" type="MPEG7MediaDescriptorType" substitutionGroup="M
ediaDescriptor"/>
   <xs:complexType name="MPEG7StatementType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Statement carrying MPEG-7 Part 5
descriptors</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element name="MPEG7ContentEntity" type="mpeg7:ContentEntityType"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MPEG7Statement" type="MPEG7StatementType" substitutionGroup="didl:Statemen
t"/>
   <!--Multimedia Representation of Bridgets-->
   <xs:complexType name="BridgetMultimediaRepresentationDescriptorType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Descriptor representing a visual or
aural representation of a Bridget</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="BridgetDescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="BridgetMultimediaRepresentationStatement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
```

```
   <xs:element name="BridgetMultimediaRepresentationDescriptor" type="BridgetMultimediaRepr
esentationDescriptorType" substitutionGroup="BridgetDescriptor"/>
   <xs:complexType name="BridgetMultimediaRepresentationStatementType">
      <xs:annotation>
         <xs:documentation>A restriction of MPEG-21 DIDL Statement carrying visual or aural
representation of a Bridget</xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element name="MultimediaRepresentation" type="mpeg7:MediaLocatorType"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetMultimediaRepresentationStatement" type="BridgetMultimediaRepre
sentationStatementType" substitutionGroup="didl:Statement"/>
   <xs:element name="TargetCondition" substitutionGroup="didl:Condition">
      <xs:annotation>
         <xs:documentation>An extension of MPEG-21 DIDL Condition carrying information
about target users of the Bridget</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:complexContent>
            <xs:extension base="didl:ConditionType">
               <xs:sequence maxOccurs="unbounded">
                  <xs:element name="RecommendedBridgetUsageCondition" type="rd:Recommendabl
eResourceType"/>
               </xs:sequence>
            </xs:extension>
         </xs:complexContent>
      </xs:complexType>
   </xs:element>
   <!--Layouting Descriptors-->
   <xs:complexType name="LayoutRefType">
      <xs:annotation>
         <xs:documentation>This complex type defines the layout presentation of a Bridget
destination</xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="BridgetLayoutUrl">
            <xs:complexType>
               <xs:simpleContent>
                  <xs:extension base="xs:anyURI">
                     <xs:attribute name="mimeType" type="xs:string"/>
                  </xs:extension>
               </xs:simpleContent>
            </xs:complexType>
         </xs:element>
         <xs:element name="BridgetLayoutScene" type="BridgetLayoutSceneType"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="MainLayoutRefType">
      <xs:annotation>
         <xs:documentation>This complex type links the bridget layout representation to the
main programme layout, where the bridget presentation takes place</xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="MainLayoutUrl" type="mlaf-pres:InlineType"/>
         <xs:element name="MainLayoutScene" type="mlaf-pres:TransformType"/>
      </xs:sequence>
```

```
   </xs:complexType>
   <xs:complexType name="BridgetLayoutSceneType">
      <xs:annotation>
         <xs:documentation>This complex type defines the XMT layouting information of a Bridget</
xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element ref="mlaf-pres:Scene"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="BridgetLayoutDescriptorType">
      <xs:complexContent>
         <xs:restriction base="BridgetDescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:element ref="BridgetLayoutStatement"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetLayoutDescriptor" type="BridgetLayoutDescriptorType" substitutionGroup="Bridg
etDescriptor"/>
   <xs:complexType name="BridgetLayoutStatementType">
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element name="BridgetLayout" type="LayoutRefType"/>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="BridgetLayoutStatement" type="BridgetLayoutStatementType" substitutionGroup="didl:St
atement"/>
   <xs:complexType name="MainLayoutDescriptorType">
      <xs:complexContent>
         <xs:restriction base="didl:DescriptorType">
            <xs:sequence>
               <xs:element ref="didmodel:Condition" minOccurs="0" maxOccurs="unbounded"/>
               <xs:choice>
                  <xs:element ref="MainLayoutStatement"/>
               </xs:choice>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MainLayoutDescriptor" type="MainLayoutDescriptorType" substitutionGroup="didl:Descri
ptor"/>
   <xs:complexType name="MainLayoutStatementType">
      <xs:complexContent>
         <xs:restriction base="didl:StatementType">
            <xs:sequence>
               <xs:element name="MainLayout">
                  <xs:complexType>
                     <xs:complexContent>
                        <xs:extension base="LayoutRefType">
                           <xs:choice>
                              <xs:element name="MainLayoutUri" type="mlaf-pres:InlineType"/>
                              <xs:element name="MainLayoutScene" type="mlaf-pres:TransformType"/>
                           </xs:choice>
                        </xs:extension>
                     </xs:complexContent>
                  </xs:complexType>
               </xs:element>
            </xs:sequence>
         </xs:restriction>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="MainLayoutStatement" type="MainLayoutStatementType" substitutionGroup="didl:Statemen
t"/>
</xs:schema>
```

Annex B
(normative)

XMTA restricted scene for MLAF presentation

The **mlaf: Scene** is a subset of **xmta:Scene** designed to simplify the scene presentation and management. The restriction refers to the set of the supported graphic primitives and their associated data types. Three XML schemas have been updated in order to support MLAF presentation and attached to this document. The update consists in taking out the elements of XMT-A (and their associated types) which are not needed in the layout description of a MLAF representation.

— **bridget_mlaf-a.xsd**, a restricted set of simple types and elements from xmt-a.xsd as defined in ISO/IEC 14496-11.

— **bridget_mlaf-od.xsd**, a restricted set of types (simpleTypes and complexTypes) and elements from xmt-od.xsd as defined in ISO/IEC 14496-11.

— **bridget_mlaf-bifs.xsd**, a restricted set of types (simpleTypes and complexTypes) and elements from xmt-bifs.xsd as defined in ISO/IEC 14496-11.

Annex C
(informative)

Bridget presentation in BIFS

The following MPEG-4 nodes are used to describe the bridget Presentation Information:

1) Appearance	17) OrderedGroup
2) AudioClip	18) PointSet
3) BitWrapper	19) PositionInterpolator2D
4) Circle	20) Rectangle
5) Color	21) ScalarInterpolator
6) Coordinate	22) Script
7) FontStyle	23) Shape
8) Group	24) Sound2D
9) ImageTexture	25) Switch
10) IndexedFaceSet	26) Text
11) IndexedLineSet	27) TextureCoordinate
12) Inline	28) TimeSensor
13) Layout	29) TouchSensor
14) LineProperties	30) Transform
15) Material2D	31) Transform2D
16) Normal	

The description of each node is provided below.

1) Appearance

```
Appearance {
        exposedField SFNode material          NULL
        exposedField SFNode texture           NULL
        exposedField SFNode textureTransform  NULL
}
```

The Appearance node specifies the visual properties of geometry by defining the material and texture nodes. The value for each of the fields in this node can be NULL. However, if the field is non-NULL, it shall contain one node of the appropriate type.

The material field, if specified, shall contain a Material node. If the material field is NULL or unspecified, lighting is off (all lights are ignored during rendering of the object that references this Appearance) and the unlit object color is (0, 0, 0).

The texture field, if specified, shall contain one of the various types of texture nodes (ImageTexture, MovieTexture or PixelTexture). If the texture node is NULL or unspecified, the object that references this Appearance is not textured.

The textureTransform field, if specified, shall contain a TextureTransform node. If the texture field is NULL or unspecified, or if the textureTransform is NULL or unspecified, the textureTransform field has no effect.

2) AudioClip

```
AudioClip {
        exposedField   SFString description      ""
        exposedField   SFBool   loop             FALSE
        exposedField   SFFloat  pitch            1.0
        exposedField   SFTime   startTime        0
        exposedField   SFTime   stopTime         0
        exposedField   MFString url              []
        eventOut       SFTime   duration_changed
        eventOut       SFBool   isActive
}
```

An AudioClip node specifies audio data that can be referenced by other nodes that require an audio source.

The description field is a textual description of the audio source. A browser is not required to display the description field but may choose to do so in addition to or in place of playing the sound.

The url field specifies the URL from which the sound is loaded. Browsers shall support at least the wavefile format in uncompressed PCM format [WAVE]. It is recommended that browsers also support the MIDI file type 1 sound format [MIDI]. MIDI files are presumed to use the General MIDI patch set. Results are not defined when the URL references unsupported data types.

The loop, startTime and stopTime exposedFields and the isActive eventOut, and their affects on the AudioClip node, are discussed in detail in ISO/IEC 14772-1:1997, 4.6.9. The "cycle" of an AudioClip is the length of time in seconds for one playing of the audio at the specified pitch.

The pitch field specifies a multiplier for the rate at which sampled sound is played. Only positive values are valid for pitch (a value of zero or less will produce undefined results). Changing the pitch field affects both the pitch and playback speed of a sound. A set_pitch event to an active AudioClip is ignored (and no pitch_changed eventOut is generated). If pitch is set to 2,0, the sound should be played one octave higher than normal and played twice as fast. For a sampled sound, the pitch field alters the sampling rate at which the sound is played. The proper implementation of the pitch control for MIDI (or other note sequence sound clip) is to multiply the tempo of the playback by the pitch value and adjust the MIDI Coarse Tune and Fine Tune controls to achieve the proper pitch change. The pitch field shall be >0,0.

A duration_changed event is sent whenever there is a new value for the "normal" duration of the clip. Typically this will only occur when the current url in use changes and the sound data have been loaded, indicating that the clip is playing a different sound source. The duration is the length of time in seconds for one cycle of the audio for a pitch set to 1,0. Changing the pitch field will not trigger a duration_changed event. A duration value of –1 implies the sound data have not yet loaded or the value is unavailable for some reason.

The isActive eventOut can be used by other nodes to determine if the clip is currently active. If an AudioClip is active, then it should be playing the sound corresponding to the sound time (i.e. in the sound's local time system with sample 0 at time 0):

fmod (now - startTime, `duration` */ pitch).*

3) BitWrapper

```
BitWrapper {
  field SFNode node NULL
  field SFInt32  type 0
  field MFString url []
  field SFString buffer ""
}
```

A node may have a dedicated node compression scheme. This compressed representation may be carried in the BIFS stream or in a separate stream.

The node field contains the node that has a compressed representation. The BitWrapper node can be used in lieu and place of the node it wraps. The type field indicates which node compression scheme shall be used, 0 being the default. It is envisioned that future node compression schemes may be developed for the same node. For this specification, AFX object code table of ISO/IEC 14496-1 defines the default schemes.

The compressed representation is carried either in a separate stream or within the scene stream. The url field indicates the stream that contains the compressed representation and the buffer field contains the compressed representation when carried within the scene. When the compressed representation is carried in separate streams by using url field, node decoders shall be configured.

In the object descriptor stream, a node decoder is indicated in the DecoderConfig descriptor for streamType 0×03, objectTypeIndication 0×05, and code defined in AFX object code table of ISO/IEC 14496-1. The decoder is configured with an AFXConfig descriptor.

Note that buffer is an array of 8-bit values. It shall not be interpreted as a UTF-8 string. For in-band scenario, compressed media stream is transmitted within a scene description stream through buffer field.

For out-band scenario, compressed media stream is transmitted outside scene description stream through the urlfield. It is used when the specific node requires upstream to send a specific information to a server.

4) Circle

```
Circle {
      exposedField SFFloat radius 1.0
}
```

This node specifies a circle centred at (0,0) in the local coordinate system. The radius field specifies the radius of the circle and shall be greater than 0. The default texture mapping coordinates are defined as the four corners of the bounding box of the circle.

5) Color

```
Color {
  exposedField MFColor color  []          # [0,1]
}
```

This node defines a set of RGB colours to be used in the fields of another node.

Color nodes are only used to specify multiple colours for a single geometric shape, such as colours for the faces or vertices of an IndexedFaceSet. A Material node is used to specify the overall material parameters of lit geometry. If both a Material node and a Color node are specified for a geometric shape, the colours shall replace the diffuse component of the material.

RGB or RGBA textures take precedence over colours; specifying both an RGB or RGBA texture and a Color node for geometric shape will result in the Color node being ignored. Details on lighting equations can be found in ISO/IEC 14772-1:1997, 4.14.

6) Coordinate

```
Coordinate {
        exposedField MFVec3f point []
}
```

This node defines a set of 3D coordinates to be used in the coord field of vertex-based geometry nodes including IndexedFaceSet, IndexedLineSet and PointSet.

7) FontStyle

```
FontStyle {
  field MFString family       "SERIF"
  field SFBool   horizontal   TRUE
  field MFString justify      "BEGIN"
  field SFString language     ""
  field SFBool   leftToRight  TRUE
  field SFFloat  size         1.0          # (0,)
  field SFFloat  spacing      1.0          # [0,)
  field SFString style        "PLAIN"
  field SFBool   topToBottom  TRUE

   }
```

The FontStyle node defines the size, family and style used for Text nodes, as well as the direction of the text strings and any language-specific rendering techniques used for non-English text. See ISO/IEC 14772-1:1997, 6.47, for a description of the Text node.

The size field specifies the nominal height, in the local coordinate system of the Text node, of glyphs rendered and determines the spacing of adjacent lines of text. Values of the size field shall be greater than zero.

The spacing field determines the line spacing between adjacent lines of text. The distance between the baseline of each line of text is (spacing × size) in the appropriate direction (depending on other fields described below). The effects of the size and spacing field are depicted in ISO/IEC 14772-1:1997, Figure 6.7 (spacing greater than 1,0). Values of the spacing field shall be non-negative.

The full description of the FontStyle element can be found in ISO/IEC 14772:1997, 6.20.

8) Group

```
Group {
        eventIn      MFNode  addChildren
        eventIn      MFNode  removeChildren
        exposedField MFNode  children       []
        field        SFVec3f bboxCenter     0 0 0
        field        SFVec3f bboxSize       -1 -1 -1
}
```

A Group node is equivalent to a Transform node, without the transformation fields.

See ISO/IEC 14772-1:1997, 4.6.5 for a description of the children, addChildren and removeChildren fields and eventIns.

The bboxCenter and bboxSize fields specify a bounding box that encloses the Group's children. This is a hint that may be used for optimization purposes. If the specified bounding box is smaller than the actual bounding box of the children at any time, then the results are undefined. A default bboxSize value, (–1 –1 –1), implies that the bounding box is not specified and if needed shall be calculated by the browser. See ISO/IEC 14772-1:1997, 4.6.4 for a description of the bboxCenter and bboxSize fields.

9) ImageTexture

```
ImageTexture {
        exposedField MFString url      []
        field        SFBool   repeatS TRUE
        field        SFBool   repeatT TRUE
}
```

The ImageTexture node defines a texture map by specifying an image file and general parameters for mapping to geometry. Texture maps are defined in a 2D coordinate system, (s, t), that ranges from 0,0 to 1,0 in both directions. The bottom edge of the image corresponds to the S-axis of the texture map and the left edge of the image corresponds to the T-axis of the texture map. The lower-left pixel of the image corresponds to s = 0, t = 0 and the top-right pixel of the image corresponds to s = 1, t = 1.

The texture is read from the URL specified by the url field. To turn off texturing, set the url field to have no values ([]). Browsers are required to support the JPEG and PNG image file formats, and in addition, may support any other image formats. Support for the GIF format [GIF] including transparent backgrounds is also recommended. See ISO/IEC 14772-1:1997, 4.5.2 for details on the url field.

Texture images may be one component (greyscale), two component (greyscale plus transparency), three component (full RGB color) or four-component (full RGB color plus transparency). An ideal VRML implementation will use the texture image to modify the diffuse color and transparency of an object's material (specified in a Material node), then perform any lighting calculations using the rest of the object's material properties with the modified diffuse color to produce the final image. The texture image modifies the diffuse color and transparency depending on how many components are in the image as follows.

a) Diffuse color is multiplied by the greyscale values in the texture image.

b) Diffuse color is multiplied by the greyscale values in the texture image; material transparency is multiplied by transparency values in texture image.

c) RGB colors in the texture image replace the material's diffuse color.

d) RGB colors in the texture image replace the material's diffuse color; transparency values in the texture image replace the material's transparency.

See ISO/IEC 14772-1:1997, 4.14 for details on lighting equations and the interaction between textures, materials and geometries.

Browsers may approximate this ideal behaviour to increase performance. One common optimization is to calculate lighting only at each vertex and combining the texture image with the color computed from lighting (performing the texturing after lighting). Another common optimization is to perform no lighting calculations at all when texturing is enabled, displaying only the colors of the texture image.

The repeatS and repeat fields specify how the texture wraps in the S and T directions. If repeatS is TRUE (the default), the texture map is repeated outside the 0-to-1 texture coordinate range in the S direction so that it fills the shape. If repeatS is FALSE, the texture coordinates are clamped in the S direction to lie within the 0-to-1 range. The repeat field is analogous to the repeatS field.

10) IndexedFaceSet

```
IndexedFaceSet {
        eventIn MFInt32 set_colorIndex
        eventIn MFInt32 set_coordIndex
        eventIn MFInt32 set_normalIndex
        eventIn MFInt32 set_texCoordIndex
        exposedField SFNode color NULL
        exposedField SFNode coord NULL
        exposedField SFNode normal NULL
        exposedField SFNode texCoord NULL
        field SFBool ccw TRUE
        field MFInt32 colorIndex []
        field SFBool colorPerVertex TRUE
        field SFBool convex TRUE
        field MFInt32 coordIndex []
        field SFFloat creaseAngle 0.0
        field MFInt32 normalIndex []
        field SFBool normalPerVertex TRUE
        field SFBool solid TRUE
        field MFInt32 texCoordIndex []
}
```

The IndexedFaceSet node represents a 3D shape formed by constructing faces (polygons) from vertices listed in the coord field. The coord field contains a Coordinate node that defines the 3D vertices referenced by the coordIndex field. IndexedFaceSet uses the indices in its coordIndex field to specify the polygonal faces by indexing into the coordinates in the Coordinate node. An index of "–1" indicates that the current face has ended and the next one begins. The last face may be (but does not have to be) followed by a "–1" index. If the greatest index in the coordIndex field is N, the Coordinate node shall contain N+1 coordinates (indexed as 0 to N). Each face of the IndexedFaceSet shall have:

a) at least three non-coincident vertices;

b) vertices that define a planar polygon;

c) vertices that define a non-self-intersecting polygon.

Otherwise, the results are undefined.

The IndexedFaceSet node is specified in the local coordinate system and is affected by the transformations of its ancestors.

Descriptions of the coord, normal and texCoord fields are provided in the Coordinate, Normal and TextureCoordinate nodes, respectively.

Details on lighting equations and the interaction between color field, normal field, textures, materials and geometries are provided in ISO/IEC 14772-1:1997, 4.14.

If the color field is not NULL, it shall contain a Color node whose colours are applied to the vertices or faces of the IndexedFaceSet as follows.

a) If colorPerVertex is FALSE, colours are applied to each face as follows:

   1) If the colorIndex field is not empty, then one colour is used for each face of the IndexedFaceSet. There shall be at least as many indices in the colorIndex field as there are faces in the IndexedFaceSet. If the greatest index in the colorIndex field is N, then there shall be N+1 colours in the Color node. The colorIndex field shall not contain any negative entries.

   2) If the colorIndex field is empty, then the colours in the Color node are applied to each face of the IndexedFaceSet in order. There shall be at least as many colours in the Color node as there are faces.

b) If colorPerVertex is TRUE, colours are applied to each vertex as follows:

   1) If the colorIndex field is not empty, then colours are applied to each vertex of the IndexedFaceSet in exactly the same manner that the coordIndex field is used to choose coordinates for each vertex from the Coordinate node. The colorIndex field shall contain at least as many indices as the coordIndex field, and shall contain end-of-face markers (–1) in exactly the same places as the coordIndex field. If the greatest index in the colorIndex field is N, then there shall be N+1 colours in the Color node.

   2) If the colorIndex field is empty, then the coordIndex field is used to choose colours from the Color node. If the greatest index in the coordIndex field is N, then there shall be N+1 colours in the Color node.

If the color field is NULL, the geometry shall be rendered normally using the Material and texture defined in the Appearance node (see ISO/IEC 14772-1:1997, 4.14 for details).

If the normal field is not NULL, it shall contain a Normal node whose normals are applied to the vertices or faces of the IndexedFaceSet in a manner exactly equivalent to that described above for applying colours to vertices/faces (where normalPerVertex corresponds to colorPerVertex and normalIndex corresponds to colorIndex). If the normal field is NULL, the browser shall automatically generate normals using creaseAngle to determine if and how normals are smoothed across shared vertices (see ISO/IEC 14772-1:1997, 4.6.3.5).

If the texCoord field is not NULL, it shall contain a TextureCoordinate node. The texture coordinates in that node are applied to the vertices of the IndexedFaceSet as follows:

a) If the texCoordIndex field is not empty, then it is used to choose texture coordinates for each vertex of the IndexedFaceSet in exactly the same manner that the coordIndex field is used to choose coordinates for each vertex from the Coordinate node. The texCoordIndex field shall contain at least as many indices as the coordIndex field and shall contain end-of-face markers (–1) in exactly the same places as the coordIndex field. If the greatest index in the texCoordIndex field is N, then there shall be N+1 texture coordinates in the TextureCoordinate node.

b) If the texCoordIndex field is empty, then the coordIndex array is used to choose texture coordinates from the TextureCoordinate node. If the greatest index in the coordIndex field is N, then there shall be N+1 texture coordinates in the TextureCoordinate node.

If the texCoord field is NULL, a default texture coordinate mapping is calculated using the local coordinate system bounding box of the shape. The longest dimension of the bounding box defines the S coordinates and the next longest defines the T coordinates. If two or all three dimensions of the bounding box are equal, ties shall be broken by choosing the X, Y or Z dimension in that order of preference. The value of the S coordinate ranges from 0 to 1, from one end of the bounding box to the other. The T coordinate ranges between 0 and the ratio of the second greatest dimension of the bounding box to the greatest dimension.

Some restrictions are specified in ISO/IEC 14496-11:2015, 7.2.2.66.2 (BIFS).

The IndexedFaceSet node represents a 3D polygon mesh formed by constructing faces (polygons) from points specified in the coord field. If the coordIndex field is not NULL, IndexedFaceSet uses the indices in its coordIndex field to specify the polygonal faces by connecting together points from the coord field. An index of –1 shall indicate that the current face has ended and the next one begins. The last face may be followed by a –1. IndexedFaceSet shall be specified in the local coordinate system and shall be affected by parent transformations. The coord field specifies the vertices of the face set and is specified by Coordinate node. If the coordIndex field is not NULL, the indices of the coordIndex field shall be used to specify the faces by connecting together points from the coord field. An index of –1 shall indicate that the current face has ended and the next one begins.

The last face may be followed by a –1. If the coordIndex field is NULL, the vertices of the coord field are laid out in their respective order to specify one face. If the color field is NULL and there is a Material node defined for the Appearance affecting this IndexedFaceSet, then the emissiveColor of the Material node shall be used to draw the faces.

In order to use 3D Mesh Coding (3DMC) with the IndexedFaceSet node, the use3DMeshCoding flag in BIFSv2Config should be set to TRUE as described in 8.5.3. This will require every IndexedFaceSet node in that elementary stream to be coded with 3DMC. Note that 3DMC does not support the use of DEF and USE within the fields of IndexedFaceSet. Also, an empty IndexedFaceSet should not be included in a stream where use3DmeshCoding flag is set to TRUE. A scene with both 3DMC coded and BIFS coded IndexedFaceSet nodes can be created by sending the compressed and uncompressed nodes in separate streams. This can be done with an Inline node or by sending separate elementary streams in the same object descriptor. The latter approach has the advantage of keeping the nodes in the same name space; see the example in 7.8 (3D Mesh Coding in BIFS scenes).

11) IndexedLineSet

```
IndexedLineSet {
        eventIn MFInt32 set_colorIndex
        eventIn MFInt32 set_coordIndex
        exposedField SFNode color NULL
        exposedField SFNode coord NULL
        field MFInt32 colorIndex []
        field SFBool colorPerVertex TRUE
        field MFInt32 coordIndex []
}
```

The IndexedLineSet node represents a 3D geometry formed by constructing polylines from 3D vertices specified in the coord field. IndexedLineSet uses the indices in its coordIndex field to specify the polylines by connecting vertices from the coord field. An index of "–1" indicates that the current polyline has ended and the next one begins. The last polyline may be (but does not have to be) followed by a "–1". IndexedLineSet is specified in the local coordinate system and is affected by the transformations of its ancestors.

The coord field specifies the 3D vertices of the line set and contains a Coordinate node.

Lines are not lit, are not texture-mapped and do not participate in collision detection. The width of lines is implementation dependent and each line segment is solid (i.e. not dashed).

If the color field is not NULL, it shall contain a Color node. The colours are applied to the line(s) as follows:

a) If colorPerVertex is FALSE:

   1) If the colorIndex field is not empty, one colour is used for each polyline of the IndexedLineSet. There shall be at least as many indices in the colorIndex field as there are polylines in the IndexedLineSet. If the greatest index in the colorIndex field is N, there shall be N+1 colours in the Color node. The colorIndex field shall not contain any negative entries.

2) If the colorIndex field is empty, the colours from the Color node are applied to each polyline of the IndexedLineSet in order. There shall be at least as many colours in the Color node as there are polylines.

b) If colorPerVertex is TRUE:

1) If the colorIndex field is not empty, colours are applied to each vertex of the IndexedLineSet in exactly the same manner that the coordIndex field is used to supply coordinates for each vertex from the Coordinate node. The colorIndex field shall contain at least as many indices as the coordIndex field and shall contain end-of-polyline markers (–1) in exactly the same places as the coordIndex field. If the greatest index in the colorIndex field is N, there shall be N+1 colours in the Color node.

2) If the colorIndex field is empty, the coordIndex field is used to choose colours from the Color node. If the greatest index in the coordIndex field is N, there shall be N+1 colours in the Color node.

If the color field is NULL and there is a Material defined for the Appearance affecting this IndexedLineSet, the emissiveColor of the Material shall be used to draw the lines. Details on lighting equations as they affect IndexedLineSet nodes are described in ISO/IEC 14772-1:1997, 4.14.

12) Inline

```
Inline {
  exposedField MFString url         []
  field        SFVec3f  bboxCenter 0 0 0      # (-,)
  field        SFVec3f  bboxSize   -1 -1 -1   # (0,) or -1, 1, -1

}
```

The Inline node is a grouping node that reads its children data from a location in the World Wide Web. Exactly when its children are read and displayed is not defined (e.g. reading the children may be delayed until the Inline node's bounding box is visible to the viewer). The url field specifies the URL containing the children. An Inline node with an empty URL does nothing.

Each specified URL shall refer to a valid VRML file that contains a list of children nodes, prototypes and routes at the top level as described in ISO/IEC 14772-1:1997, 4.6.5. The results are undefined if the URL refers to a file that is not VRML or if the VRML file contains non-children nodes at the top level.

If multiple URLs are specified, the browser may display a URL of a lower preference VRML file while it is obtaining, or if it is unable to obtain, the higher preference VRML file. Details on the url field and preference order can be found in ISO/IEC 14772-1:1997, 4.5.

The results are undefined if the contents of the URL change after it has been loaded.

The bboxCenter and bboxSize fields specify a bounding box that encloses the Inline node's children. This is a hint that may be used for optimization purposes. The results are undefined if the specified bounding box is smaller than the actual bounding box of the children at any time. A default bboxSize value, (–1, –1, –1), implies that the bounding box is not specified and if needed shall be calculated by the browser. A description of the bboxCenter and bboxSize fields is in ISO/IEC 14772-1:1997, 4.6.4.

13) Layout

```
Layout {
      eventIn MFNode addChildren
      eventIn MFNode removeChildren
      exposedField MFNode children []
      exposedField SFBool wrap FALSE
      exposedField SFVec2f size -1, -1
      exposedField SFBool horizontal TRUE
      exposedField MFString justify ["BEGIN"]
      exposedField SFBool leftToRight TRUE
      exposedField SFBool topToBottom TRUE
      exposedField SFFloat spacing 1.0
      exposedField SFBool smoothScroll FALSE
      exposedField SFBool loop FALSE
      exposedField SFBool scrollVertical TRUE
      exposedField SFFloat scrollRate 0.0
      exposedField SFInt32 scrollMode 0
}
```

The Layout node specifies the placement (layout) of its children in various alignment modes as specified. For text children, this is by their fontStyle fields, and for non-text children by the fields horizontal, justify, leftToRight, topToBottom and spacing present in this node. It also provides the functionality of scrolling its children horizontally or vertically.

The children field shall specify a list of nodes that are to be arranged. Note that the children's position is implicit and that order is important.

The wrap field specifies whether children are allowed to wrap to the next row (or column in vertical alignment cases) after the edge of the layout frame is reached. If wrap is set to TRUE, children that would be positioned across or past the frame boundary are wrapped (vertically or horizontally) to the next row or column. If wrap is set to FALSE, children are placed in a single row or column that is clipped if it is larger than the layout.

When wrap is TRUE, if text objects larger than the layout frame need to be placed, these texts shall be broken down into pieces that are smaller than the layout. The preferred places for breaking text are spaces, tabs, hyphens, carriage returns and line feeds. When there is no such character in the texts to be broken, the texts shall be broken at the last character that is entirely placed in the layout frame.

The size field specifies the width and height of the layout frame.

The horizontal, justify, leftToRight, topToBottom and spacing fields have the same meaning as in the FontStyle node (see ISO 14496-11:2015, 7.2.2.61).

The scrollRate field specifies the time needed in seconds to scroll the layout in the given direction. For example, a layout of 200 × 100 pixels scrolling vertically with a scrollRate value of 2 will translate its objects vertically of 100/2 times the simulation frame duration in seconds (e.g. 1,65 pixels at 30 fps). When scrollRate is zero, then there is no scrolling and the remaining scroll-related fields are ignored.

The smoothScroll field selects between smooth and line-by-line/character-by-character scrolling of children. When TRUE, smooth scroll is applied.

The loop field specifies continuous looping of children when set to TRUE. When loop is FALSE, child nodes that have scrolled out of the scroll layout frame will be deleted. When loop is TRUE, then the set of children scrolls continuously, wrapping around when they have scrolled out of the layout area. If the set of children is smaller than the layout area, some empty space will be scrolled with the children. If the set of children is bigger than the layout area, then only some of the children will be displayed at any point in time. When scrollVertical is TRUE and loop is TRUE and scrollRate is negative (top-to-bottom scrolling), then the bottom-most object will reappear on top of the layout frame as soon as the topmost object has scrolled entirely into the layout frame.

The scrollVertical field specifies whether the scrolling is done vertically or horizontally. When set to TRUE, the scrolling rate shall be interpreted as a vertical scrolling rate and a positive rate shall be

interpreted as scrolling towards the top. When set to FALSE, the scrolling rate shall be interpreted as a horizontal scrolling rate and a positive rate shall mean scrolling to the right.

Objects are placed one by one, in the order they are given in the children list. Text objects are placed according to the horizontal, justify, leftToRight, topToBottom and spacing fields of their FontStyle node. Other objects are placed according to the same fields of the Layout node. The reference point for the placement of an object is the reference point as left by the placement of the previous object in the list.

In the case of vertical alignment, objects may be placed with respect to their top, bottom, center or baseline. The baseline of non-text objects is the same as their bottom.

Spacing shall be coherent only within sequences of objects with the same orientation (same value of horizontal field). The notions of top edge, bottom edge, base line, vertical center, left edge, right edge, horizontal center, line height and row width shall have a single meaning over coherent sequences of objects. This means that over a sequence of objects where horizontal is TRUE, topToBottom is TRUE and spacing has the same value, then the vertical size of the lines is computed as follows:

— maxAscent is the maximum of the ascent on all text objects;

— maxDescent is the maximum of the descent on all text objects;

— maxHeight is the maximum height of non-text objects.

If the minor mode in the justify field of the layout is FIRST (baseline alignment), then the non-text objects shall be aligned on the baseline, which means the vertical size of the line is:

size = max(maxAscent, maxHeight) + maxDescent

If the minor mode in the justify field of the layout is any other value, then the non-text objects shall be aligned with respect to the top, bottom or center, which means the size of the line is:

size = max(maxAscent+maxDescent, maxHeight)

The first line is placed with its top edge flush to the top edge of the layout; the base line is placed maxAscent units lower and the bottom edge is placed maxDescent units lower. The center line is in the middle, between the top and bottom edges. The top edges of subsequent lines are placed at regular intervals of value spacing × size.

The other cases can be inferred from the above description. When the orientation is vertical, then the baseline, ascent and descent are not useful for the computation of the width of the rows. All objects only have a width. Column size is the maximum width over all objects.

14) LineProperties

```
LineProperties {
      exposedField SFColor lineColor 0, 0, 0
      exposedField SFInt32 lineStyle 0
      exposedField SFFloat width 1.0
}
```

The LineProperties node specifies line parameters used in 2D and 3D rendering.

The lineColor field specifies the colour with which to draw the lines and outlines of 2D geometries.

The lineStyle field shall contain the line style type to apply to lines. The allowed values are:

lineStyle	Description
0	Solid
1	Dash
2	Dot

3	Dash-dot
4	Dash-dash-dot
5	Dash-dot-dot

The terminal shall draw each line style in a manner that is distiguishable from each other line style.

The width field determines the width, in the local coordinate system, of rendered lines. The width is not subject to the local transformation.

The cap and join style to be used are as follows. The wide lines should end with a square form flush with the end of the lines.

15) Material2D

```
Material2D {
      exposedField SFColor emissiveColor 0.8, 0.8, 0.8
      exposedField SFBool filled FALSE
      exposedField SFNode lineProps NULL
      exposedField SFFloat transparency 0.0
}
```

The Material2D node specifies the characteristics of a rendered 2D Shape. Material2D shall be used as the material field of an Appearance node in certain circumstances (see ISO 14496-11:2015, 7.2.2.6.2).

The emissiveColor field specifies the colour of the 2D Shape. If the shape is not filled, the interior is not drawn.

The filled field specifies whether rendered nodes are filled or drawn using lines. This field affects IndexedFaceSet2D, Circle and Rectangle nodes. If the rendered node is not filled, the line shall be drawn centred on the rendered node outline. That means that half the line will fall inside the rendered node and the other half outside.

The lineProps field contains information about line rendering in the form of a LineProperties node. When filled is true, if lineProps is null, no outline is drawn; if lineProps is non-null, an outline is drawn using lineProps information. When filled is false and lineProps is null, an outline is drawn with default width (1), default style (solid) and as line colour the emissive color of the Material2D. When filled is false and lineProps is defined, line color, width and style are taken from the lineProps node. See ISO 14496-11:2015, 7.2.2.75 for more information on LineProperties.

The transparency field specifies the transparency of the 2D Shape and applies both to the filled interior as well as to the outline when drawn.

The part of the line which lies outside of the geometry shall not be sensitive to pointer activity.

When mapping texture onto a geometry and an outline is to be drawn, the texture shall first mapped onto the geometry, where the geometry dimensions are those without an outline. Then after the geometry is textured, the outline shall be drawn.

16) Normal

```
Normal {
        exposedField MFVec3f vector []
}
```

This node defines a set of 3D surface normal vectors to be used in the vector field of some geometry nodes (e.g. IndexedFaceSet and ElevationGrid). This node contains one multiple-valued field that contains the normal vectors. Normals shall be of unit length.

17) OrderedGroup

```
OrderedGroup {
      eventIn MFNode addChildren
      eventIn MFNode removeChildren
      exposedField MFNode children []
```

```
      exposedField MFFloat order []
}
```

The OrderedGroup node controls the visual layering order of its children. When used as a child of a Layer2D node, it allows the control of which shapes obscure others. When used as a child of a Layer3D node, it allows content creators to specify the rendering order of elements of the scene that have identical z values. This allows conflicts between coplanar or close polygons to be resolved.

The addChildren eventIn specifies a list of objects that shall be added to the OrderedGroup node.

The removeChildren eventIn specifies a list of objects that shall be removed from the OrderedGroup node.

The children field is the current list of objects contained in the OrderedGroup node.

When the order field is empty (the default), children are layered in order, first child to last child, with the last child being rendered last. If the order field contains values, one value is assigned to each child. Entries in the order field array match the child in the corresponding element of the children field array. The child with the lowest order value is rendered before all others. The remaining children are rendered in increasing order. The child corresponding to the highest order value is rendered last. If there are more children than entries in the order field, those children that do not have a drawing order are drawn in the order in which they appear in the children field, but after the ones that have an entry in the order field.

If there are more order entries than children, the excess order entries are ignored.

Since 2D shapes have no z value, this is the sole determinant of the visual ordering of the shapes. However, when the OrderedGroup node is used with 3D shapes, its ordering mechanism shall be used in place of the natural z order of the shapes themselves. The resultant image shall show the shape with the highest order value on top, regardless of its z value. However, the resultant z-buffer contains a z value corresponding to the shape closest to the viewer at that pixel.

The order shall be used to specify which geometry should be drawn first to avoid conflicts between coplanar or close polygons.

Content authors shall use this functionality carefully since, depending on the Viewpoint, 3D shapes behind a given object in the natural z order may appear in front of this object.

18) PointSet

```
PointSet {
  exposedField  SFNode  color     NULL
  exposedField  SFNode  coord     NULL
}
```

The PointSet node specifies a set of 3D points, in the local coordinate system, with associated colours at each point. The coord field specifies a Coordinate (see ISO/IEC 14772-1:1997, 6.12) node (or instance of a Coordinate node). The results are undefined if the coord field specifies any other type of node. PointSet uses the coordinates in order. If the coord field is NULL, the point set is considered empty.

PointSet nodes are not lit, not texture-mapped, nor do they participate in collision detection. The size of each point is implementation dependent.

If the color field is not NULL, it shall specify a Color (see ISO/IEC 14772-1:1997, 6.9) node that contains at least the number of points contained in the coord node. The results are undefined if the color field specifies any other type of node. Colours shall be applied to each point in order. The results are undefined if the number of values in the Color node is less than the number of values specified in the Coordinate node.

If the color field is NULL and there is a Material node defined for the Appearance node affecting this PointSet node, the emissiveColor of the Material node shall be used to draw the points. More details on lighting equations can be found in ISO/IEC 14772-1:1997, 4.14.

19) PositionInterpolator2D

```
PositionInterpolator2D {
      eventIn SFFloat set_fraction
      exposedField MFFloat key []
      exposedField MFVec2f keyValue []
      eventOut SFVec2f value_changed
}
```

This node linearly interpolates among a set of SFVec2f values. This is appropriate for interpolating a translation. The vectors are interpreted as absolute positions in object space. The keyValue field shall contain exactly as many values as in the key field.

20) Rectangle

```
Rectangle {
      exposedField SFVec2f size 2, 2
}
```

This node specifies a rectangle centred at (0,0) in the local coordinate system. The size field specifies the horizontal and vertical size of the rendered rectangle.

21) ScalarInterpolator

```
ScalarInterpolator {
        eventIn      SFFloat set_fraction
        exposedField MFFloat key           []
        exposedField MFFloat keyValue      []
        eventOut     SFFloat value_changed
}
```

This node linearly interpolates among a set of SFFloat values. This interpolator is appropriate for any parameter defined using a single floating point value, e.g. width, radius, intensity, etc. The keyValue field shall contain exactly as many numbers as there are keyframes in the key field.

22) Script

```
Script {
  exposedField MFString url             []
  field        SFBool   directOutput    FALSE
  field        SFBool   mustEvaluate    FALSE
  # And any number of:
  eventIn      eventTypeName eventName
  field        fieldTypeName fieldName initialValue
  eventOut     eventTypeName eventName
}
```

The Script node is used to program behaviour in a scene. Script nodes typically receive events that signify a change or user action, contain a program module that performs some computation and effect change somewhere else in the scene by sending output events. Each Script node has associated programming language code, referenced by the url field, that is executed to carry out the Script node's function. That code will be referred to as "the script" in the rest of this description.

Browsers are not required to support any specific language. See ISO/IEC 14772-1:1997, 4.12 for detailed information on scripting languages. Browsers are required to adhere to the language bindings of languages specified in annexes of the specification. See ISO/IEC 14772-1:1997, 4.5.2 for details on the url field.

When the script is created, any language-dependent or user-defined initialization is performed. The script is able to receive and process events that are sent to it. Each event that can be received shall be declared in the Script node using the same syntax as is used in a prototype definition:

— *eventIn type name*

The type can be any of the standard VRML fields and name shall be an identifier that is unique for this Script node.

The Script node should be able to generate events in response to the incoming events. Each event that can be generated shall be declared in the Script node using the following syntax:

— *eventOut type name*

Script nodes cannot have exposedFields. The implementation ramifications of exposedFields is far too complex and thus not allowed.

If the Script node's mustEvaluate field is FALSE, the browser can delay sending input events to the script until its outputs are needed by the browser. If the mustEvaluate field is TRUE, the browser should send input events to the script as soon as possible, regardless of whether the outputs are needed. The mustEvaluate field should be set to TRUE only if the Script has effects that are not known to the browser (such as sending information across the network); otherwise, poor performance may result.

Once the script has access to a VRML node (via an SFNode or MFNode value either in one of the Script node's fields or passed in as an eventIn), the script should be able to read the contents of that node's exposed field. If the Script node's directOutput field is TRUE, the script may also send events directly to any node to which it has access and may dynamically establish or break routes. If directOutput is FALSE (the default), then the script may only affect the rest of the world via events sent through its eventOuts.

A script is able to communicate directly with the VRML browser to get the current time, the current world URL and so on. This is strictly defined by the API for the specific language being used.

It is expected that all other functionality (such as networking capabilities, multi-threading capabilities and so on) will be provided by the scripting language.

The location of the Script node in the scene graph has no effect on its operation. For example, if a parent of a Script node is a Switch node with whichChoice set to –1 (i.e. ignore its children), the Script continues to operate as specified (receives and sends events).

23) Shape

```
Shape {
      exposedField SFNode appearance NULL
      exposedField SFNode geometry   NULL
}
```

The Shape node has two fields: appearance and geometry which are used to create rendered objects in the world. The appearance field specifies an Appearance node that specifies the visual attributes (e.g. material and texture) to be applied to the geometry. The geometry field specifies a geometry node. The specified geometry node is rendered with the specified appearance nodes applied.

If the geometry field is NULL, the object is not drawn.

24) Sound2D

```
Sound2D {
      exposedField SFFloat intensity 1.0
      exposedField SFVec2f location 0,0
      exposedField SFNode source NULL
      field  SFBool spatialize TRUE
}
```

The Sound2D node relates an audio BIFS sub-graph to the other parts of a 2D audiovisual scene. It shall not be used in 3D contexts (see ISO 14496-11:2015, 7.1.1.2.1). By using this node, sound may be attached to a group of visual nodes. By using the functionality of the audio BIFS nodes, sounds in an audio scene may be filtered and mixed before being spatially composed into the scene.

The intensity field adjusts the loudness of the sound. Its value ranges from 0,0 to 1,0 and this value specifies a factor that is used during the playback of the sound.

The location field specifies the location of the sound in the 2D scene.

The source field connects the audio source to the Sound2D node.