DRAFT AMENDMENT

ISO/IEC 14496-16:2011/DAM 3

ISO/IEC JTC **1**/SC **29** Secretariat: **JISC**

Voting begins on:
2016-01-04

Voting terminates on:
2016-04-04

Information technology — Coding of audio-visual objects —

Part 16:
Animation Framework eXtension (AFX)

AMENDMENT 3: Printing material and 3D graphics coding for browsers

Technologies de l'information — Codage des objets audiovisuels —

Partie 16: Extension du cadre d'animation (AFX)

AMENDEMENT 3: .

ICS: 35.040

Reference number
ISO/IEC 14496-16:2011/DAM 3:2015(E)

COPYRIGHT PROTECTED DOCUMENT

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

Foreword

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

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

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

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

Amendment 1 to ISO/IEC 14496-16:2011 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects — Part 16: Animation Framework eXtension (AFX), AMENDMENT 3: Printing material and 3D graphics coding for browsers

In Section 4.3.6.4.2, replace :

```
IndexedRegionSet {
  coord Coordinate {point [
    0 0 0, 1 0 0, 1 1 0, 0 1 0, 0 1 1, 0 0 1, 1 0 1, 1 1 1, 0.5 0.5 0
  ]}
  texCoord TextureCoordinate {point [
    0 0, 0.333 0, 0.667 0, 1 0, 0 1, 0.333 1, 0.667 1, 1 1, 0.667 0.5
  ]}
  region [
    Region {
      coordIndex [
        2 1 8 -1, 1 0 8 -1, 0 3 8 -1, 3 2 8 -1, 1 2 7 -1, 7 6 1 -1, 5 6 7 -1, 7 4 5 -1
      ]
      texCoordIndex [5 1 0 4 7 6 2 3 8]
      texture ImageTexture {url "../pix/136.png"}
    }
    Region {
      coordIndex [4 3 0 -1, 0 5 4 -1, 6 5 0 -1, 0 1 6 -1, 3 4 7 -1, 7 2 3 -1]
      texCoordIndex [4 7 6 5 1 0 3 2]
      texture ImageTexture {url "../pix/245.png"}
    }
  ]
}
```

With:

```
IndexedRegionSet {
[…]
    Region {
      coordIndex [
        2 1 8 -1, 1 0 8 -1, 0 3 8 -1, 3 2 8 -1, 6 5 0 -1, 0 1 6 -1, 5 6 7 -1, 7 4 5 -1
      ]
[…]
    }
    Region {
      coordIndex [4 3 0 -1, 0 5 4 -1, 1 2 7 -1, 7 6 1 -1, 3 4 7 -1, 7 2 3 -1]
[…]
    }
}
```

Before 4.3.7 (which shall be renumbered to 4.3.8), "Solid representation", add:

4.3.7 3D meshes with Printing Material Texture

4.3.7.1 Introduction

The **IndexedPrintingRegionSet (IPRS)** node is based on the **IndexedRegionSet** described in 4.3.6 and it describes region based printing material information which can be used in the 3D printing application. The main design concept of **IPRS** is that the representation shall be easy to the designer. Based on this concept, **IPRS** has adopted texture mapping method for describing the printing material information because it is very

popular to designer. Thanks to the region based texture mapping, per face or region or global material mapping is possible.

4.3.7.2 IndexedPrintingRegionSet node

4.3.7.2.1 Node interface

```
IndexedPrintingRegionISet {
    eventIn       MFInt32   set_colorIndex
    eventIn       MFInt32   set_coordIndex
    eventIn       MFInt32   set_normalIndex
    eventIn       MFInt32   set_texCoordIndex
    exposedField  SFString  unit                      #mm, cm, m, inch
    exposedField  SFFloat   minimumVerticalResolution #unit is unit
    exposedField  SFNode    color                     NULL
    exposedField  SFNode    coord                     NULL
    exposedField  SFNode    normal                    NULL
    exposedField  SFNode    texCoord                  NULL
    field         SFBool    ccw                       TRUE
    field         MFInt32   colorIndex                []      # [-1,inf)
    field         SFBool    colorPerVertex            TRUE
    field         SFBool    convex                    TRUE
    field         MFNode    printingRegions           []
    field         SFFloat   creaseAngle               0       # [0,inf)
    field         MFInt32   normalIndex               []      # [-1,inf)
    field         SFBool    normalPerVertex           TRUE
    field         SFBool    solid                     TRUE
    field         MFInt32   texCoordIndex             []      # [0,inf)
}
```

4.3.7.2.2 Functionality and semantics

An **IPRS** node has exactly the same fields as an **IRS** one, except for the physical size information of printout, and ***region*** field, which has been replaced by ***printingMaterialRegion***.

4.3.7.3 PrintingRegion node

4.3.7.3.1 Node interface

```
PrintingRegion {
    exposedField  MFInt32   printingMaterialType          NULL
    exposedField  SFFloat   surfaceThickness              #unit is unit
    exposedField  SFNode    color                         NULL
    exposedField  SFNode    normal                        NULL
    exposedField  SFNode    texCoord                      NULL
    exposedField  SFNode    colorTexture                  NULL
    exposedField  SFNode    printingMaterialTexture       NULL
    exposedField  SFNode    textureTransform              NULL
    field         MFInt32   colorIndex                    []      # [-1,inf)
    field         MFInt32   coordIndex                    []      # [-1,inf)
    field         MFInt32   normalIndex                   []      # [-1,inf)
    field         MFInt32   texCoordIndex                 []      # [0,inf)
    field         MFInt32   printingMaterialTexCoordIndex []      # [0,inf)
}
```

4.3.7.3.2 Functionality and semantics

field name	semantic
unit	It has single string value which shall be one of the length unit defined in the ISO/IEC 23005-6 (UnitTypeCS). It defines the unit of the ***coord*** field. When this value is set as mm, all the ***coord*** values are interpreted as mm (milli-meter).
surfaceThickness	It defines the suggested surface thickness of the model. For example a cylinder model with 5 ***surfaceThickness*** is interpreted as "Print the cylinder with 5 mm surface and leave the inside empty." Here the unit for ***surfaceThickness*** is same as ***unit*** files.
minimumVerticalResolution	It defined the resolution for the acquisition process. When the model is designed based on the scanner, the scanner resolution is mentioned here. When this field is set as 0.1, the 3D printing resolution which is smaller than ***minimumVerticalResolution*** does not make sense. In this way, the printing resolution may be estimated. Here the unit for ***minimumVerticalResolution*** is same as ***unit*** files.
printingRegions	It has **PrintingRegion** node which defined the physical material characteristics of print-out. It has multiple nodes. When single element is provided, the whole model is considered as one region.
printingMaterialType	It has multiple integers that defines the printing materials of print-out as a reference to a classification defined in ISO/IEC 23005-6 (PrintingMaterialCS). When a single material is provided, the whole region is printed as one material.
colorTexture	It defines the color texture used in each region for rendering purpose.
printingMaterialTexture	It defines the printing material texture map used in each region. It shall be lossless gray image format such as PNG because the lossy compression results in a misunderstanding on the printing material information. And the values in the image are restricted by ***printingMaterialType.*** All values in the printing material texture are one value in the ***printingMaterialType*** array. For example, When the ***printingMaterialType*** is provided as [0, 1, 2], the ***printingMaterialTexture*** shall has values among 0, 1, and 2. When the array length of ***printingMaterialType*** is 1, the ***printingMaterialTexture*** is null because this region is printed as single material with ***printingMaterialType***.
texCoordIndex	It defines the texture coordinate indexes of ***colorTexture***.
printingMaterialTexCoordIndex	It defines the texture coordinate indexes of ***printingMaterialTexture***. When this is null, ***texCoordIndex*** is used as ***printingMaterialTexCoordIndex.***

4.3.7.4 Examples

The following examples are based on the IRS representation described in 4.3.6.4.3 Third IRS example. It has two **printingRegion** and each **printingRegion** has ***colorTexture*** and ***printignMaterialTexture*** as illustrated in Figure Amd3.X.

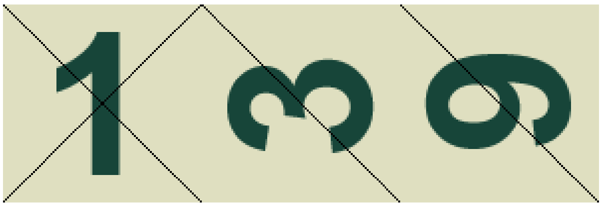

Figure Amd3.X-(a)

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	2	1	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	2	2	1	1	0	0	0	0
0	0	0	1	1	1	0	0	0	2	2	2	1	1	1	0	0	0
0	0	0	0	1	1	0	0	2	2	2	2	1	1	1	1	0	0
0	0	0	0	0	1	0	2	2	2	2	2	1	1	1	1	1	0

Figure Amd3.X-b

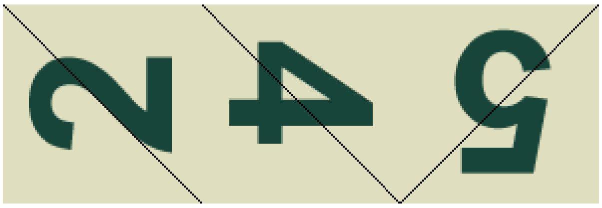

Figure Amd3.X-c

2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	1	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	2	2	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0

Figure Amd3.X-d

Figure Amd3.X — IPRS example with two printingRegion;
(a) colorTexture(136.png), (b) printingMaterialTexture(136_material.png)
(c) colorTexture(245.png), (d) printingmaterialTexture(245_material.png

In this example, three material [0, 1, 2] are used thus the values in the printingMaterialTexture are restricted by 0, 1 and 2.

```
IndexedPrintingRegionSet{
  unit “mm”
  minimumVerticalResolution 0.1
  coord Coordinate {
    point [ 0 0 0, 1 0 0, 1 1 0, 0 1 0, 0 1 1, 0 0 1, 1 0 1, 1 1 1, 0.5 0.5 0 ]
  }
  texCoord TextureCoordinate {
    point [ 0 0, 0.333 0, 0.667 0, 1 0, 0 1, 0.333 1, 0.667 1, 1 1, 0.1667 0.5 ]
  }
  printingRegions [
    PrintingRegion {
      surfaceThickness 10.0
      printingMaterialType [0, 1, 2]
      coordIndex [ 2 1 8 -1, 1 0 8 -1, 0 3 8 -1, 3 2 8 -1, 6 5 0 -1, 0 1 6 -1, 5 6 7 -1,
7 4 5 -1 ]
      texCoordIndex [5 1 0 4 7 6 2 3 8]
      colorTexture            ImageTexture  {  url "../pix/136.png"  }
      printingMaterialTexture ImageTexture  {  url "../material/136_material.png" }
    }
    PrintingRegion {
      surfaceThickness 15.0
       printingMaterialType [0, 1, 2]
      coordIndex [4 3 0 -1, 0 5 4 -1, 1 2 7 -1, 7 6 1 -1, 3 4 7 -1, 7 2 3 -1]
      texCoordIndex [4 7 6 5 1 0 3 2]
      colorTexture             ImageTexture   { url "../pix/245.png" }
       printingMaterialTexture  ImageTexture   { url "../material/245_material.png" }
    }
  ]
}
```

Before 5 (which shall be renumbered to 6), "Web3D Coding", add:

6 Web3D Coding

6.1 Introduction

This document describes a scene representation using a JSON schema. The reason for choosing JSON is the native support by modern browsers and easiness of integration with javascript. The JSON schema is used to connect the object graph elements, MPEG AFX tools, images and shaders. The main components of the library (Figure X) are the JSON parser and the GraphicsCodec, whereas the GraphicsCodec contains both the SC3DMC and BBA decoders. The input of the library is the JSON description file. The JSON Parser analyses its data and calls the appropriate decoders, the SC3DMC decoder for the mesh data and BBA decoder for the animated data. The output of the decoders is then used to initialize the corresponding glIndexedFaceSet structures that are then used by the WebGL engine to render the scene.

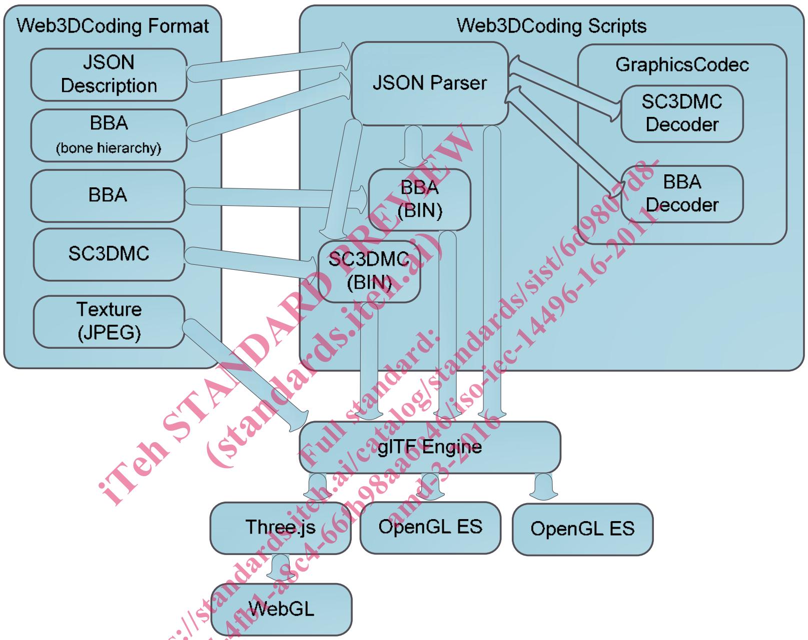

Figure X. Overview of the Web3DCoding architecture

6.2 Scope

Define a JSON schema that allows connections between object graph elements to MPEG 4 AFX compression tools (SC3DMC and BBA), image compression tools (JPEG, JPEG2000 and PNG) and shaders.

6.3 JSON Schema

```
{
  "type": "object",
  "properties": {
    "object": {
        "type": "object",
        "properties": {
            "name": {
                 "type": "string",
                 "description" : "The name of the object. The name has to be unique in the object
                 definition"
            },

            "shapes": {
                "type": "array",
                "items": {
                     "type": "object",
                     "properties" : {
                         "geometry_filename" : {
                             "type" : "string",
```