International Telecommunication Union

ITU-R

Radiocommunication Sector of ITU

# Recommendation ITU-R BS.2076-2

**(10/2019)**

# Audio definition model

**BS Series**

**Broadcasting service (sound)**

International Telecommunication Union

## Foreword

The role of the Radiocommunication Sector is to ensure the rational, equitable, efficient and economical use of the radio-frequency spectrum by all radiocommunication services, including satellite services, and carry out studies without limit of frequency range on the basis of which Recommendations are adopted.

The regulatory and policy functions of the Radiocommunication Sector are performed by World and Regional Radiocommunication Conferences and Radiocommunication Assemblies supported by Study Groups.

## Policy on Intellectual Property Right (IPR)

ITU-R policy on IPR is described in the Common Patent Policy for ITU-T/ITU-R/ISO/IEC referenced in Resolution ITU-R 1. Forms to be used for the submission of patent statements and licensing declarations by patent holders are available from https://www.itu.int/ITU-R/go/patents/en where the Guidelines for Implementation of the Common Patent Policy for ITU-T/ITU-R/ISO/IEC and the ITU-R patent information database can also be found.

### Series of ITU-R Recommendations

(Also available online at http://www.itu.int/publ/R-REC/en)

| Series | Title |
|---|---|
| **BO** | Satellite delivery |
| **BR** | Recording for production, archival and play-out; film for television |
| **BS** | **Broadcasting service (sound)** |
| **BT** | Broadcasting service (television) |
| **F** | Fixed service |
| **M** | Mobile, radiodetermination, amateur and related satellite services |
| **P** | Radiowave propagation |
| **RA** | Radio astronomy |
| **RS** | Remote sensing systems |
| **S** | Fixed-satellite service |
| **SA** | Space applications and meteorology |
| **SF** | Frequency sharing and coordination between fixed-satellite and fixed service systems |
| **SM** | Spectrum management |
| **SNG** | Satellite news gathering |
| **TF** | Time signals and frequency standards emissions |
| **V** | Vocabulary and related subjects |

***Note***: *This ITU-R Recommendation was approved in English under the procedure detailed in Resolution ITU-R 1.*

*Electronic Publication*
Geneva, 2019

# RECOMMENDATION ITU-R BS.2076-2*

## Audio definition model

(2015-2017-2019)

**Scope**

This Recommendation describes the structure of a metadata model that allows the format and content of audio files to be reliably described. This model, called the Audio Definition Model (ADM), specifies how XML metadata can be generated to provide the definitions of tracks in an audio file.

**Keywords**

ADM, Audio Definition Model, BW64, Metadata, Wave-file, WAVE, object-based, channel-based, scene-based, renderer, XML, XSD, format, immersive

The ITU Radiocommunication Assembly,

*considering*

*a)* that Recommendation ITU-R BS.2051 – Advanced sound system for programme production, highlights the need for a file format that is capable of dealing with the requirements for future audio systems;

*b)* that Recommendation ITU-R BS.1909 – Performance requirements for an advanced multichannel stereophonic sound system for use with or without accompanying picture, outlines the requirements for an advanced multichannel stereophonic sound system;

*c)* that it is desirable that there is a single open standard for a metadata model for defining audio content that file and streaming formats could either adopt or become compatible with by means of suitable interfacing,

*recommends*

for the following use cases:

– applications requiring a generic metadata model for, and a formalized description of, custom/proprietary audio formats and content (including codecs);

– generating and parsing audio metadata with general-purpose tools, such as text editors;

– an organization’s internal production developments, where multi-purpose metadata needs to be added;

– a human-readable and hand-editable file for describing audio configurations (such as describing a mixing studio channel configuration) in a consistent and translatable format is needed,

to use the Audio Definition Model (ADM) described in Annex 1 for metadata to describe audio formats used in programme production and international exchange.

* Radiocommunication Study Group 6 made editorial amendments to this Recommendation in February 2020 in accordance with Resolution ITU-R 1.

TABLE OF CONTENTS

*Page*

Annex 1 – Audio Definition Model ..... 4

Overview of changes in this edition ..... 4

1 Introduction ..... 6

2 Background ..... 6

2.1 Cooking analogy ..... 6

2.2 Brief overview ..... 7

3 Description of the model ..... 7

3.1 Format ..... 9

3.2 Content ..... 10

4 Common definitions ..... 11

5 ADM elements ..... 11

5.1 audioTrackFormat ..... 11

5.2 audioStreamFormat ..... 12

5.3 audioChannelFormat ..... 13

5.4 audioBlockFormat ..... 15

5.5 audioPackFormat ..... 26

5.6 audioObject ..... 31

5.7 audioContent ..... 37

5.8 audioProgramme ..... 40

5.9 audioTrackUID ..... 45

5.10 audioFormatExtended ..... 46

5.11 Time parameters format ..... 47

6 Use of IDs ..... 47

7 <chna> Chunk ..... 49

8 Coordinate system ..... 50

9 Common parameter descriptions for all typeDefinitions ..... 52

9.1 gain ..... 52

9.2 importance ..... 52

9.3 headLocked ..... 53

*Page*

9.4 headphoneVirtualise ..... 53

10 Parameter descriptions for typeDefinition of ‘Objects’ ..... 53

10.1 diffuse ..... 54

10.2 channelLock ..... 54

10.3 jumpPosition and interpolationLength ..... 54

10.4 zoneExclusion ..... 56

10.5 objectDivergence ..... 56

10.6 screenRef and audioProgrammeReferenceScreen ..... 57

11 Parameter descriptions for typeDefinition of ‘HOA’ ..... 58

11.1 order and degree ..... 58

11.2 normalization ..... 59

11.3 nfcRefDist ..... 60

11.4 screenRef ..... 60

11.5 Ambisonics Channel Numbering ..... 60

12 Relationship and application of gain parameters in the ADM ..... 60

13 Application of position-related parameters in the ADM ..... 62

14 References ..... 63

Annex 2 (informative) – Examples of ADM usage ..... 63

1 Channel-based example ..... 63

1.1 Summary of elements ..... 63

1.2 Element Relationships ..... 64

1.3 Sample code ..... 65

2 Object-based example ..... 68

2.1 Summary of elements ..... 68

2.2 Element Relationships ..... 68

2.3 Sample code ..... 69

3 Scene-based example ..... 71

3.1 Summary of elements ..... 71

3.2 Element Relationships ..... 72

3.3 Sample code ..... 73

*Page*

4 Material exchange format mapping example ..... 76
4.1 Summary of elements ..... 76
4.2 Element Relationships ..... 77
4.3 Sample code ..... 78
5 Personalized audio example ..... 82
5.1 Summary of elements ..... 82
5.2 Element Relationships ..... 84
5.3 Sample code ..... 85
6 22.2 multichannel programme with an alternative dialogue example ..... 92
6.1 Summary of elements ..... 92
6.2 Element relationships ..... 96
6.3 Sample code ..... 97
7 Example of the use of the Matrix type ..... 111
7.1 Summary of elements ..... 111
7.2 Element Relationships ..... 112
7.3 Sample code ..... 112

# Annex 1

# Audio Definition Model

**Overview of changes in this edition**

This overview provides a list of updates and changes to this Recommendation from previous version(s).

i) Editorial changes and additional clarifying text / examples to ensure clear understanding, describing existing features in BS.2076-1:

| Item | Description | Section(s) |
|---|---|---|
| 1 | Addition of clarification text for nested audioObjects and their timing parameters | 5.6.7 |
| 2 | Addition of clarification text with respect to the <chna> chunk | 7 |
| 3 | Clarification of the usage of rtime and duration<br>Addition of a section about time formats | 5.4.1<br>5.11 |
| 4 | The description of the use of equation parameter for HOA content has been revised | 5.4.3.4 |
| 5 | Renumbering of sections, tables and figures, correction of references | several |

ii) Technical corrections and additional clarification text:

| Item | Description | Section(s) |
|---|---|---|
| 6 | Addition of missing default values for several elements and attributes | 5.4.3.2, 5.6.1, 10.5 |
| 7 | Correction of obligation/optionality of some elements and attributes (e.g. the ADM version attribute in audioFormatExtended is now mandatory); addition of clarifying text on how to handle cases where some optional elements are not present and how to handle old ADM files defined according to previous versions | 5.1.2, 5.2.2, 5.9.2, 5.10.1 |
| 8 | Clarification of handling of dynamic metadata and interpolation | 5.4.3.3, 10.3 |
| 9 | Clarification on the use of typeDefinition="Matrix" (clarify that input and output formats are not restricted to 'DirectSpeakers' type) | 5.4.3.2, 5.5.4 |
| 10 | Correction of the values of the default screen size with respect to the screenRef element | 10.6 |
| 11 | Addition of text about the definition of a default audioProgramme | 5.8 |
| 12 | Correction of minimum and maximum values for some elements, e.g. maximum distance of conditioned channelLock | 5.4.3.4, 5.5.5 |
| 13 | Clarification of the rules of common HOA parameters in audioPackFormat and audioBlockFormat | 5.4.3.4, 5.5.5 |
| 14 | Clarification of application of attributes and elements for different coordinate systems | 5.4.3.3, 5.6.2, 5.8.3 |
| 15 | Clarification on position and extent parameters | 5.4.3.3 |

iii) Further technical changes including the definition of new elements and attributes:

| Item | Description | Section(s) |
|---|---|---|
| 16 | Start time and duration of audioBlockFormat, audioObject and audioProgramme can now also be defined in audio samples | 5.4.1, 5.6.1, 5.8.1 |
| 17 | Gain and importance sub-elements of audioBlockFormat are now available for all typeDefinitions | 5.4.3 |
| 18 | An additional gain element, mute element, and positionOffset element have been added to audioObject | 5.6.2 |
| 19 | A headLocked element is introduced that allows definition if the position of audio content should be influenced by headtracking/tracking data. | 5.4.3, 5.6.2 |
| 20 | A headphoneVirtualise element is introduced that allows to bypass headphone rendering for specific elements and contains an attribute to define the Direct-To-Reverberant Ratio (DRR) | 5.4.3 |
| 21 | A new sub-element alternativeValueSet is introduced to allow adjustment of audioObject parameters when specific content is chosen for playback | 5.6.5 |
| 22 | Gain values can now be given either as linear values or in dB | 5.4.3, 5.6.2 |

| Item | Description | Section(s) |
|---|---|---|
| 23 | It is now possible to define labels in multiple languages | 5.6.2, 5.7.2, 5.8.2 |
| 24 | Dynamic metadata for typeDefinition="Matrix" has been added | 5.4.3.2 |
| 25 | An authoringInformation element has been added to describe the production reference layout and the renderers used during monitoring | 5.8.2, 5.8.6 |

iv) Further additional explanatory text, describing old and new features of BS.2076:

| Item | Description | Section(s) |
|---|---|---|
| 26 | Explanatory text about the combination of the different gain-related parameters to a combined playback gain and gain interaction | 12 |
| 27 | Explanatory text about the application of position-related parameters and position interaction | 13 |
| 28 | Parameters which are common to all typeDefinitions are now described in § 9 | 9 |

## 1 Introduction

Audio for broadcasting and cinema is evolving towards an immersive and interactive experience, which requires the use of more flexible audio formats. A fixed channel-based approach is not sufficient to encompass these developments and so combinations of channel, object and scene-based formats are being developed. Report ITU-R BS.2266 [1] and Recommendations ITU-R BS.1909 [2] and ITU-R BS.2051 [3] highlight these developments and the need for the production chain to accommodate them.

The central requirement for allowing all the different types of audio to be distributed, whether by file or by streaming, is that whatever file/stream format is used, metadata should co-exist to fully describe the audio. Each individual track within a file or stream should be able to be correctly rendered, processed or distributed according to the accompanying metadata. To ensure compatibility across all systems, the Audio Definition Model is an open standard that will make this possible.

## 2 Background

The purpose of this model is to formalise the description of audio. It is not a format for carrying audio. This distinction will help in the understanding of the model.

### 2.1 Cooking analogy

To help explain what the ADM actually does, it may be useful to consider a cooking analogy. The recipe for a cake will contain a list of ingredients, instructions on how to combine those ingredients and how to bake the cake.

The ADM is like a set of rules for writing the list of ingredients; it gives a clear description of each item, for example: 2 eggs, 400g flour, 200g butter, 200g sugar.

The ADM provides the instructions for combining ingredients but does not tell you how to do the mixing or baking; in the audio world that is what the renderer does.

The ADM is in general compatible with wave-file based formats such as the BW64 format specified in Recommendation ITU-R BS.2088 [7], the BWF as defined by the ITU in [4] and other wave based formats that support the usage of the needed additional chunks.

When used in the context of Recommendation ITU-R BS.2088 file, the <*chna*> chunk of the BS.2088 file is like the bar code on the packet of each of the ingredients; this code allows us to look up the model's description of each item. The bag containing the actual ingredients is like the 'data' chunk of the BS.2088 file that contains the audio samples.

From a Recommendation ITU-R BS.2088 file point of view, one would look at the bar codes on each ingredient in the bag, and use that to look up the description of each item in the bag. Each description follows the structure of the model. There might be ingredients such as breadcrumbs, which could be divided into its own components (flour, yeast, etc.); which is like having an audio object containing multiple channels (e.g. 'stereo' containing 'left' and 'right').

## 2.2 Brief overview

This model will initially use XML as its specification language, though it could be mapped to other languages such as JSON (JavaScript Object Notation) if required. When it is used with Recommendation ITU-R BS.2088 files, the XML can be embedded in specific chunks, for example the <axml> chunk, of the file.

The model is divided into two sections, the **content** part, and the **format** part. The content part describes what is contained in the audio, so will describe things like the language of any dialogue, the loudness and so on.

The format part describes the technical nature of the audio so it can be decoded or rendered correctly. Some of the format elements may be defined before having any audio signals, whereas the content parts can usually only be completed after the signals have been generated.

While this model is based around a wave-file based format, it is a more general model. However, examples are given using Recommendation ITU-R BS.2088 according to the definition in [7] as this explains more clearly how the model works. It is also expected that the model's parameters are added to in subsequent versions of this specification to reflect the progress in audio technology.

# 3 Description of the model

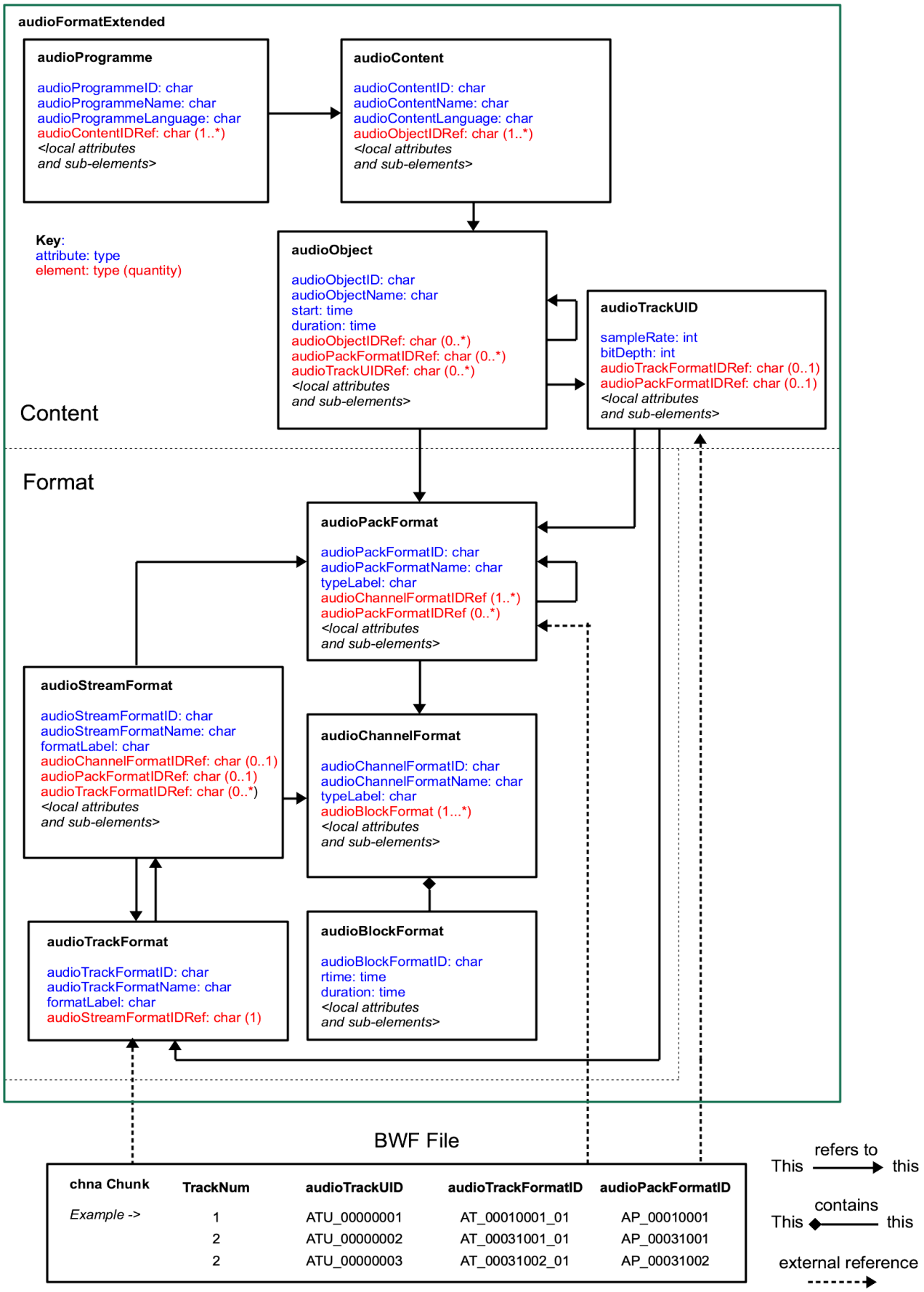
The overall diagram of the model is given in Fig. 1. This shows how the elements relate to each other and illustrates the split between the content and format parts. It also shows the <*chna*> chunk of a BS.2088 file and how it connects the tracks in the file to the model.

Where a BS.2088 file contains a number of audio tracks, it is necessary to know what each track is. The <*chna*> chunk contains a list of numbers corresponding to each track in the file. Hence, for a 6-track file, the list is at least 6 long. For each track there is an audioTrackFormatID number and an audioTrackUID number (notice the additional 'U' which stands for 'unique'). The reason the list could be longer than the number of tracks is that a single track may have different definitions at different times so will require multiple audioTrackUIDs and references.

The audioTrackFormatID is used to look up the definition of the format of that particular track. The audioTrackFormatIDs are not unique; for example, if a file contains five stereo pairs, there will be five identical audioTrackFormatIDs to describe the 'left' channel, and five to describe the 'right' channel. Thus, only two different audioTrackFormatIDs will need to be defined. However, audioTrackUIDs are unique (hence the 'U'), and they are there to uniquely identify the track. This use of IDs means that the tracks can be ordered in any way in the file; their IDs reveal what those tracks are.

FIGURE 1

**Overall UML Model**



### 3.1 Format

The audioTrackFormatID answers the question “What is the format of this track?” The audioTrackFormat will also contain an audioStreamFormatID, which allows identification of the combination of the audioTrackFormat and audioStreamFormat. An audioStreamFormat describes a decodable signal.

The audioStreamFormat is made up of one or more audioTrackFormats. Hence, the combination of audioStreamFormat and audioTrackFormat reveals whether the signal has to be decoded or not.

The next stage is to find out what type of audio the stream is; for example, it may be a conventional channel (e.g. ‘front left’), an audio object (e.g. something named ‘guitar’ positioned at the front), a HOA (Higher Order Ambisonics) component (e.g. ‘X’) or a group of channels. Inside audioStreamFormat there will be a reference to either an audioChannelFormat or audioPackFormat that will describe the audio stream. There will only be one of these references.

If audioStreamFormat contains an audioChannelFormat reference (i.e. audioChannelFormatIDRef) then audioStreamFormat is one of several different types of audioChannelFormat. An audioChannelFormat is a description of a single waveform of audio. In audioChannelFormat there is a typeDefinition attribute, which is used to define what the type of channel is.

The typeDefinition attribute can be set to ‘DirectSpeakers’, ‘HOA’, ‘Matrix’ ‘Objects’ or ‘Binaural’. For each of those types, there is a different set of sub-elements to specify the static parameters associated with that type of audioChannelFormat. For example, the ‘DirectSpeakers’ type of channel has the sub-element ‘speakerLabel’ for allocating a loudspeaker to the channel.

To allow audioChannelFormat to describe dynamic channels (i.e. channels that change in some way over time), it uses audioBlockFormat to divide the channel along the time axis. The audioBlockFormat element will contain a start time (relative to the start time of the parent audioObject) and duration. Within audioBlockFormat there are time-dependent parameters that describe the channel which depend upon the audioChannelFormat type.

For example, the ‘Objects’ type of channel has the sub-elements ‘azimuth’, ‘elevation’ and ‘distance’ to describe the location of the sound. The number and duration of audioBlockFormats is not limited, there could be an audioBlockFormat for every sample if something moves rapidly, though that might be a bit excessive! At least one audioBlockFormat is required and so static channels will have one audioBlockFormat containing the channel’s parameters.

If audioStreamFormat refers to an audioPackFormat, it describes a group of channels. An audioPackFormat element groups together one or more audioChannelFormats that belong together (e.g. a stereo pair). This is important when rendering the audio, as channels within the group may need to interact with each other.

The reference to an audioPackFormat containing multiple audioChannelFormats from an audioStreamFormat usually occurs when the audioStreamFormat contains non-PCM audio which carries several channels encoded together. AudioPackFormat would usually not be referred from audioStreamFormat for most channel and scene-based formats with PCM audio. Where this reference does exist, the function of audioPackFormat is to combine audioChannelFormats that belong together for rendering purposes.

For example, ‘stereo’, ‘5.1’, ‘1st order Ambisonics’ would all be examples of an audioPackFormat. Note that audioPackFormat just describes the format of the audio. For example, a file containing 5 stereo pairs will contain only one audioPackFormat to describe ‘stereo’. It is possible to nest audioPackFormats; a ‘2nd order HOA’ could contain a ‘1st order HOA’ audioPackFormat alongside audioChannelFormats for the R, S, T, U & V components.

### 3.2 Content

Using an audio scene with five stereo pairs as an example, the audioTrackFormat defines which audio tracks are left and right, not which ones belong together, nor what is represented in them. AudioObject is used to determine which tracks belong together and where they are in the file. This element links the actual audio data with the format, and this is where audioTrackUID comes in.

For a stereo pair (in PCM), audioObject will contain references to two audioTrackUIDs; therefore, those two tracks will contain stereo audio. It will also contain a reference to audioPackFormat, which defines the format of those two tracks as a stereo pair.

As there are five stereo pairs in this example, 5 audioObject elements will be needed. Each one will contain the same reference to a stereo audioPackFormat, but will contain different reference to audioTrackUIDs, as each stereo pair is carrying different audio. The order of audioTrackUIDRefs is not important in an audioObject, as the format definition through audioTrack, audioStreamFormat, audioChannelFormat and audioPackFormat determines which track is which.

The audioObject element also contains start and duration attributes. This start time is the time when the signal for the object starts in a file or recording. Thus, if start is "00:00:10.00000", the signal for the object will start 10 seconds into the track in the audio file.

As audioPackFormat can be nested, it follows that audioObjects can be nested. Therefore, the audioObject will contain not only references to the two audioTrackUIDs carrying the stream, but also references to two audioObjects, one for the 5.1 and one for the 2.0.

AudioObject is referred to by audioContent, which gives a description of the content of the audio; it has parameters such as language (if there is dialogue) and the loudness parameters. Some of the values for these parameters can only be calculated after the audio has been generated, and this is why they are not in the format part.

AudioProgramme brings all the audioContent together; it combines them to make the complete 'mix'.

For example:

– an audioProgramme may contain audioContent for 'narrator' and another one for 'background music';
– an audioProgramme for France may contain audioContents called 'dialogue-fr' and 'backgroundMusic', and another audioProgramme for the UK which contains audioContents called 'dialogue-en' and the same 'backgroundMusic'.

Multiple audioProgramme elements can be defined in one ADM XML tree representation. This facilitates the description of a presentation that represents a predefined number of meaningful mixes that users can choose from. Each audioProgramme element may reference just a subset of audioContent elements of the ADM XML tree. This is one method to enable the ADM to describe personalized audio.

For example:

– Following the previous example for audioProgramme, a single ADM XML tree can contain both French and English audioProgramme elements.
– An ADM XML tree describing a sports program can contain audioProgramme elements for a home team and an away team. The home team audioProgamme may contain audioContent elements for a 'home team biased commentary', and another one for 'ambience'. The away team audioProgramme may contain audioContent for an 'away team biased commentary' and the same 'ambience'.

TABLE 1

**Alternative mixes**

| | Ambience | Neutral commentary | Home team biased commentary | Away team biased commentary |
|---|---|---|---|---|
| Default mix | • | • | | |
| Home team | • | | • | |
| Away team | • | | | • |

## 4 Common definitions

For many situations, particularly in channel and scene-based work, many of the required formats will be common. For example, mono, stereo and 5.1 all have common definitions and it would be inefficient to generate and carry a mass of XML every time one of these formats needs to be described. Common definitions are specified in Recommendation ITU-R BS.2094 [8].

This set is available in Recommendation ITU-R BS.2094 [8] as an attached XML file. This reference file will not have to be included in a file using the ADM but can be externally referred to. Therefore, a file will not need to carry the XML of the format if only common definitions are used. The occasions any ADM XML code will need to be carried in a file is when audioProgramme, audioContent and audioObject are used, or custom definitions are required.

## 5 ADM elements

Each of the elements within the ADM is described in the following subsections.

### 5.1 audioTrackFormat

The audioTrackFormat element corresponds to a single set of samples or data in a single track in a storage medium. It is used to describe what format the data is in, allowing a renderer to decode the signal correctly. It is referred from the audioStreamFormat element, which is used to identify the combination of tracks required to decode the track data successfully.

For PCM audio an audioStreamFormat will refer to a single audioTrackFormat and so the two elements are effectively describing the same thing. In this case, both the audioTrackFormat and the audioStreamFormat may be omitted. Then the audioTrackUID has to refer to the corresponding audioChannelFormat and the same number is used for the ‘yyyyxxxx’ parts of AT_yyyyxxxx_zz, AS_yyyyxxxx and AC_yyyyxxxx. For coded audio, multiple audioTrackFormats will have to be combined in a single audioStreamFormat to generate decodable data.

Software that parses the model can start from either audioTrackFormat or audioStreamFormat. To allow for this flexibility audioTrackFormat can also refer back to the audioStreamFormat.

If the audioStreamFormat references an audioTrackFormat then the audioTrackFormat shall refer back to the same audioStreamFormat.

### 5.1.1 Attributes

TABLE 2

**AudioTrackFormat attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| audioTrackFormatID | ID for track, see § 6. The yyyy digits of AT_yyyyxxxx_nn_represent the type of audio contained in the track. The yyyyxxxx digits should match the audioStreamFormat yyyyxxxx digits | AT_00010001_01 | Yes |
| audioTrackFormatName | Name for track | PCM_FrontLeft | Yes |
| formatLabel | Descriptor of the format | 0001 | Optional |
| formatDefinition | Description of the format | PCM | Optional |

### 5.1.2 Sub-elements

TABLE 3

**audioTrackFormat sub-elements**

| Sub-element | Description | Example | Quantity |
|---|---|---|---|
| audioStreamFormatIDRef | Reference to an audioStreamFormat | AS_00010001 | 1 (see Note below) |

NOTE – Earlier versions (Recommendations ITU-R BS.2076-0 and ITU-R BS.2076-1) of this Recommendation specified the above quantity as “0 or 1”, but this was an error and not the original intention. As some existing ADM files (based on Recommendations ITU-R BS.2076-0 or ITU-R BS.2076-1) may lack this sub-element due to this error, any software that reads ADM files should tolerate audioStreamFormatIDRef being absent. However, any new software should now always include this sub-element when generating ADM files.

### 5.1.3 Sample code

```
<audioTrackFormat                                    audioTrackFormatID="AT_00010001_01"
audioTrackFormatName="PCM_FrontLeft" formatDefinition="PCM" formatLabel="0001">
  <audioStreamFormatIDRef>AS_00010001</audioStreamFormatIDRef>
</audioTrackFormat>
```

## 5.2 audioStreamFormat

A stream is a combination of tracks (or one track) required to render a channel, object, HOA component or pack. The audioStreamFormat establishes a relationship between audioTrackFormats and the audioChannelFormats or audioPackFormat. Its main use is to deal with non-PCM encoded tracks, where one or more audioTrackFormats must be combined to represent a decodable signal that covers several audioChannelFormats (by referencing an audioPackFormat). For PCM audio, an audioStreamFormat will refer to a single audioTrackFormat and a single audioChannelFormat. In this case, both the audioStreamFormat and the audioTrackFormat may be omitted. Then the audioTrackUID has to refer to the corresponding audioChannelFormat and the same number is used for the ‘yyyyxxxx’ parts of AT_yyyyxxxx_zz, AS_yyyyxxxx and AC_yyyyxxxx.

### 5.2.1 Attributes

TABLE 4

**audioStreamFormat attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| audioStreamFormatID | ID for the stream, see § 6. The yyyy digits of AS_yyyyxxxx_represent the type of audio contained in the stream. The xxxx digits should match the audioChannelFormat xxxx digits. | AS_00010001 | Yes |
| audioStreamFormatName | Name of the stream | PCM_FrontLeft | Yes |
| formatLabel | Descriptor of the format | 0001 | Optional |
| formatDefinition | Description of the format | PCM | Optional |

### 5.2.2 Sub-elements

TABLE 5

**audioStreamFormat sub-elements**

| Sub-element | Description | Example | Quantity |
|---|---|---|---|
| audioChannelFormatIDRef | Reference to audioChannelFormat | AC_00010001 | 0 or 1 |
| audioPackFormatIDRef | Reference to audioPackFormat | AP_00010003 | 0 or 1 |
| audioTrackFormatIDRef | Reference to audioTrackFormat | AT_00010001_01 | 0…* (see Note below) |

NOTE – Earlier versions (Recommendations ITU-R BS.2076-0 and ITU-R BS.2076-1) of this Recommendation specified this quantity as "1", but this was an error and not the original intention. Any new software that reads ADM files should be aware that some existing ADM files (based on Recommendations ITU-R BS.2076-0 or ITU-R BS.2076-1) may only have the audioTrackFormatIDRef sub-element within audioStreamFormat, but may lack the audioStreamFormatIDRef sub-element within audioTrackFormat (see § 5.1.2).

Only one of audioPackFormatIDRef or audioChannelFormatIDRef can be used, not both in the same element.

### 5.2.3 Sample code

```
<audioStreamFormat audioStreamFormatID="AS_00010001"
audioStreamFormatName="PCM_FrontLeft" formatDefinition="PCM"
formatLabel="0001">
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
</audioStreamFormat>
```

## 5.3 audioChannelFormat

An audioChannelFormat represents a single sequence of audio samples on which some action may be performed, such as movement of an object, which is rendered in a scene. It is sub-divided in the time domain into one or more audioBlockFormats.

### 5.3.1 Attributes

TABLE 6

**audioChannelFormat attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| audioChannelFormatName | Name of the channel | FrontLeft | Yes |
| audioChannelFormatID | ID of the channel, see § 6 for the use of the audioChannelFormatID in typical channel configurations. The yyyy digits of AC_yyyyxxxx_represent the type of audio contained in the channel. The xxxx digits should match the audioStreamFormat xxxx digits. | AC_00010001 | Yes |
| typeLabel | Descriptor of the type of channel | 0001 | Optional * |
| typeDefinition | Description of the type of channel | DirectSpeakers | Optional * |

* At least one of typeLabel or typeDefinition is required.

The typeDefinition of the audioChannel Format specifies the type of audio it is describing, and also determines which parameters are used within its audioBlockFormat children.
Currently, there are five different typeDefinitions:

TABLE 7

**typeDefinitions**

| typeDefinition | typeLabel | Description |
|---|---|---|
| DirectSpeakers | 0001 | For channel-based audio, where each channel feeds a speaker directly |
| Matrix | 0002 | For all other typeDefinitions, where signals are matrixed together, such as Mid-Side, Lt/Rt |
| Objects | 0003 | For object-based audio where channels represent audio objects (or parts of objects), so include positional information |
| HOA | 0004 | For scene-based audio where Ambisonics and HOA are used |
| Binaural | 0005 | For binaural audio, where playback is over headphones |
| User Custom | 1yyy to Fyyy | For user custom types. |

### 5.3.2 Sub-elements

TABLE 8

**audioChannelFormat sub-elements**

| Sub-element | Description | Attributes | Quantity |
|---|---|---|---|
| audioBlockFormat | Time division of channel containing dynamic metadata | See § 5.4 | 1...* |
| frequency | Describes the high and/or low cut-off frequency for the audio in Hz | typeDefinition = “lowPass" or “highPass" | 0...2 |

The optional frequency parameter allows a frequency range of the audio to be described. This can be either low-pass or high-pass, or by combining both to achieve band-pass and band-stop. The most common use of this is for LFE channels where a low-pass frequency limit (e.g. 200 Hz) can be described.

### 5.3.3 Sample code

```
<audioChannelFormat audioChannelFormatID="AC_00010001"
audioChannelFormatName="FrontLeft" typeDefinition="DirectSpeakers">
  <audioBlockFormat ...>
    ...
  </audioBlockFormat>
</audioChannelFormat>
```

## 5.4 audioBlockFormat

An audioBlockFormat represents a single sequence of audioChannelFormat samples with fixed parameters, including position, within a specified time interval.

### 5.4.1 Attributes

TABLE 9

**audioBlockFormat attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| audioBlockFormatID | ID for block | AB_00010001_00000001 | Yes |
| rtime | Start time of block (relative to the start time of the parent audioObject) The start time is in the time format as described in § 5.11. | 00:00:00.00000 or 00:00:00.00000S48000 | Optional Default when not present: 00:00:00.00000 |
| duration | Duration of block. The duration is in the time format as described in § 5.11. | 00:00:05.00000 or 00:00:05.00000S48000 | Optional Default when not present: Unbounded duration |

The last 8 hexadecimal digits in the audioBlockFormatID contain the index for the block within the channel, starting at 00000001 for the first block.

If *rtime* is not used then the block starts at 00:00:00.00000. If *duration* is not used then the block lasts for the whole duration of the channel.

If there is only one audioBlockFormat within an audioChannelFormat, the characteristics of the parent audioChannelFormat are considered to be static over time, therefore *rtime* and *duration* should be omitted. When there is more than one audioBlockFormat within an audioChannelFormat, the characteristics of the parent audioChannelFormat are assumed to be dynamic over time, therefore both *rtime* and *duration* should be used.

Most of the sub-elements within audioBlockFormat are dependent upon the typeDefinition or typeLabel of the parent audioChannelFormat element.

Time restrictions imposed by the audioObject element apply to both dynamic and static metadata regardless of typeDefinitions. Currently, there are five different defined typeDefinitions:

TABLE 10

**typeDefinitions**

| typeDefinition | typeLabel | Description |
|---|---|---|
| DirectSpeakers | 0001 | For channel-based audio, where each channel feeds a speaker directly |
| Matrix | 0002 | For all other typeDefinitions, where signals are matrixed together, such as Mid-Side, Lt/Rt |
| Objects | 0003 | For object-based audio where channels represent audio objects (or parts of objects) and so include positional information |
| HOA | 0004 | For scene-based audio where Ambisonics and HOA are used |
| Binaural | 0005 | For binaural audio, where playback is over headphones |
| User Custom | 1yyy to Fyyy | For user custom types. |

### 5.4.2 Sample code

```
<audioBlockFormat audioBlockFormatID="AB_00010001_00000001" rtime="00:00:00.00000"
duration="00:00:05.00000">
  ...
</audioBlockFormat>
```

### 5.4.3 Sub-elements

TABLE 11

**Common audioBlockFormat sub-elements**

| Sub-element | Attribute | Description | Units | Example | Quantity | Default |
|---|---|---|---|---|---|---|
| gain | gainUnit | Definition of a gain value to be applied to all audio samples corresponding to the audioBlockFormat. An optional gainUnit attribute (either ‘linear’ or ‘dB’) can be used to define the unit of the gain value. The default unit is ‘linear’. For a detailed description of the application of this gain value see § 12. | gain value, default is linear value | 0.5 (linear), −6 (dB) | 0 or 1 | 1.0 |
| importance | | Importance of the audioChannelFormat, defined for the duration of the current audioBlockFormat. | 0 to 10 | 10 | 0 or 1 | 10 |
| headLocked | | Indicates if the perceived location of the audio element is locked to the head (flag = 1) or not locked (flag = 0) See § 9.3 | 0/1 flag | 1 | 0 or 1 | 0 |

TABLE 11 (*end*)

| Sub-element | Attribute | Description | Units | Example | Quantity | Default |
|---|---|---|---|---|---|---|
| headphoneVirtualise | bypass | Specifies whether the object should be virtualised using a headphone virtualiser or not (1=renderer to stereo, 0=renderer with headphone virtualiser) See § 9.4 | 1/0 flag | 1 | 0 or 1 | 0 |
| | DRR | Direct to Reverberant Ratio in dB. See § 9.4 | dB | -130..130 | 0 or 1 | 130 (anechoic-all direct) |

#### 5.4.3.1 If audioChannelFormat.typeDefinition == "DirectSpeakers"

For channel-based systems, this is the metadata used to describe the channel. If the channel is intended to be played out through a specific loudspeaker, then use *speakerLabel* to indicate the label of that speaker. While both the maximum and minimum values for the three position elements are available (using the bound attribute), they should be avoided, as the exact position should normally be specified by omitting the *bound* attribute.

TABLE 12

**audioBlockFormat sub-elements for DirectSpeakers**

| Sub-element | Attribute | Bound attribute | Description | Units/ Values | Example | Quantity |
|---|---|---|---|---|---|---|
| speakerLabel | | N/A | A reference to the label of the speaker position | – | M-30 | 0...* |
| position | coordinate="azimuth" | | Exact azimuth location of sound | Degrees | −30.0 | 1 |
| position | coordinate="azimuth" | max | Max. azimuth location of sound | Degrees | −22.5 | 0 or 1 |
| position | coordinate="azimuth" | min | Min. azimuth location of sound | Degrees | −30.0 | 0 or 1 |
| position | coordinate="elevation" | | Exact elevation location of sound | Degrees | 0.0 | 1 |
| position | coordinate="elevation" | max | Max. elevation location of sound | Degrees | 5.0 | 0 or 1 |
| position | coordinate="elevation" | min | Min. elevation location of sound | Degrees | 0.0 | 0 or 1 |
| position | coordinate="distance" | | Exact normalized distance from origin | Normalized to 1 | 1.0 | 0 or 1 |
| position | coordinate="distance" | max | Max. normalized distance from origin | Normalized to 1 | 0.8 | 0 or 1 |

TABLE 12 (*end*)

| Sub-element | Attribute | Bound attribute | Description | Units/ Values | Example | Quantity |
|---|---|---|---|---|---|---|
| position | coordinate="distance" | min | Min. normalized distance from origin | Normalized to 1 | 0.9 | 0 or 1 |
| position | screenEdgeLock | | Defines a speaker position at a screen edge | Left, right, top, bottom | Left | 0 … 2 |

The **screenEdgeLock** attribute allows a speaker to be positioned on the edge of the screen. The attribute can be used in combination with the coordinate="elevation" and/or the coordinate="azimuth" attribute and it is set to a string stating at which edge of the screen to the speaker position should be assumed to be (if screen-size information is available), so it is either "left", "right", "top", "bottom". The coordinate attribute must still be included so it is clear which dimension is being set, and to provide an alternative position should the screen not exist or no screen-size information be available.

The example XML code below illustrates how a speaker positioned on the right edge of the screen can be defined (with an alternative position of −29.0 degrees should the screen not exist).

```
<audioBlockFormat ...>
  <speakerLabel>M-SC</speakerLabel>
  <position coordinate="azimuth" screenEdgeLock="right">-29.0</position>
  <position coordinate="elevation">0.0</position>
  <position coordinate="distance">1.0</position>
</audioBlockFormat>
```

If two screenEdgeLock positions are required (for corners of the screen) then the two position ADM elements must be used as shown in the example below. This is because XML does not allow multiple attributes of the same name within the same element.

```
  <position coordinate="azimuth" screenEdgeLock="right">-29.0</position>
  <position coordinate="elevation" screenEdgeLock="top">15.0</position>
```

The distance measure is normalized, but an absolute reference distance is available in audioPackFormat. These coordinates are based on the polar system, as this is the common way of describing channel and speaker locations. However, it is also possible to use the Cartesian coordinate system by using different coordinate attributes ('X', 'Y' and 'Z'); and this system is described in more detail in § 8.

#### 5.4.3.1.1 Sample code

```
<audioBlockFormat ...>
  <speakerLabel>M-30</speakerLabel>
  <position coordinate="azimuth">-30.0</position>
  <position coordinate="elevation">0.0</position>
  <position coordinate="distance">1.0</position>
</audioBlockFormat>
```

### 5.4.3.2 If audioChannelFormat.typeDefinition == "Matrix"

This is for matrix channels, such as mid-side and Lt/Rt. The matrix element contains a list of coefficient sub-elements which each refer to other channels and a multiplication factor. All the matrix coefficients in this list should be added together to generate the matrix equation.

There are three types of matrix that can be defined: encoding, decoding and direct:

– An encoding matrix will be typically used to describe how the audio signals have been encoded to generate matrixed audio signals.

– A decoding matrix will be typically used to describe how the audio signals can be converted from matrixed audio signals to another type of output (typically, but not restricted to the "DirectSpeakers" typeDefinition). This could be the reverse process of the encoding matrix. The encoding matrix can reference a decoding matrix to connect related matrices.

– A direct matrix can convert between the same typeDefinition channels (e.g. channel-based to channel-based) directly (such as channel-based downmixing).

The audioPackFormat (see § 5.5.4) contains sub-elements that group Matrix channels and allow cross-referencing between encoding and decoding matrices.

For example, the encoding matrix element of a 'Side' channel will contain two matrix coefficient sub-elements, one with the value 0.5 referring to "Left" and the other with a value of −0.5 referring to 'Right'; this gives Side=0.5*Left-0.5*Right.

An example of a decoding matrix would be Left=0.5*Mid+0.5*Side, where 'Left' becomes a channel-based output.

A direct matrix example would be a 5.1->LoRo downmix where
Lo=Left+0.7071*Centre+0.7071*LeftSurround &
Ro=Right+0.7071*Centre+0.7071*RightSurround.

The values for gain and phase shift can either be constants (using gain and phase) or they may be variables (using gainVar and phaseVar) that allow the renderer to decide the value, maybe via another source of metadata.

TABLE 13

**audioBlockFormat sub-elements for Matrix**

| Sub-element | Attribute | Description | Quantity | Default |
|---|---|---|---|---|
| outputChannelFormatIDRef* | – | For defining a decoding or direct matrix, this is the output audioChannelFormat that defines the channel being decoded to. | 0 or 1 | |
| jumpPosition | | If jumpPosition is set to 1 the position will change instantly from the previous block's position. If set to 0 then interpolation of the position will take the entire length of the block. | 0 or 1 | 0 |
| | interpolationLength | If the interpolationLength attribute is used, and the jumpPosition value is 1, then the interpolation will take as long as the specified value. The interpolation length should be shorter or equal than the block's duration and it should be specified in seconds (at least 5 d.p). | 0 or 1 | Duration of block |
| matrix | – | See Table 14. | 1 | |

* This element name has been editorially changed from *outputChannelIDRef*, which was the incorrect name used in the original version of BS.2076-1. Therefore, ADM parsing software should be aware that *outputChannelIDRef* might occur in some files instead of *audioChannelFormatIDRef* and should be able to read both.

TABLE 14

**matrix sub-elements**

| Sub-element | Attribute | Description | Units | Example | Quantity | Default |
|---|---|---|---|---|---|---|
| coefficient | gainUnit | Unit for attribute of ‘gain’. If gainUnit is not used, ‘linear’ unit is assumed. | | linear / dB | 0 or 1 | ‘linear’ |
| coefficient | gain | Multiplication factor of another channel. Constant value. Type: float | Linear or logarithmic gain value* | −0.5 | 0...* Note: no more than one use of each attribute can be specified. | 1.0 |
| coefficient | gainVar | Multiplication factor of another channel. Variable. Type: string (reference to float) | A variable representing a linear gain value* | clev | | - |
| coefficient | phase | Phase shift of another channel. Constant value. Type: float | degrees | 90 | | 0 |
| coefficient | phaseVar | Phase shift of another channel. Variable. Type: string (reference to float) | A variable representing a value in degrees | ph | | - |
| coefficient | delay | Time delay of another channel. Constant value Type: float | ms (float) | 10.5 | | 0.0 |
| coefficient | delayVar | Time delay of another channel. Variable Type: string (reference to float) | A variable representing a time in ms | del | | - |
| coefficient | | Reference to another audioChannelFormat ID | | AC_00010001 | 1...* | |

* A negative linear gain value implies an inversion of the signal.

#### 5.4.3.2.1 Sample code

```
<audioBlockFormat ...>
  <outputChannelIDRef>AC_00010001</outputChannelIDRef>
  <jumpPosition  interpolationLength="0.50000">1</jumpPosition>
  <matrix>
    <coefficient gain="0.5">AC_00021001</coefficient>
    <coefficient gain="0.5">AC_00021002</coefficient>
  </matrix>
</audioBlockFormat>
```

#### 5.4.3.3 If audioChannelFormat.typeDefinition == “Objects”

This is for object-based audio where the position of the audio object may change dynamically. As well as the polar coordinates of the object, there are parameters for the object’s size, and whether it is a diffuse or coherent sound.

The channelLock parameter will inform a renderer to send the object’s audio to the nearest speaker or channel, rather than the usual panning, interpolation, etc. The jumpPosition parameter will ensure the renderer can control the temporal interpolation of the position values, so the object will move in space in the time specified by the interpolationLength attribute, rather than move smoothly to the next position over the whole duration of the block.

The position elements use the coordinate attribute to specify which axis is used. The primary coordinate system is the Polar coordinate system, which uses azimuth, elevation and distance axes. However, it is possible to specify other axes for other coordinates such as X, Y and Z for the Cartesian coordinate system. This is described in more detail in § 8.

The position and object size parameters definitions depend upon the coordinate system used, so they are each described in Tables 15 and 16.

For a polar/spherical coordinate system:

TABLE 15

**audioBlockFormat sub-elements for Objects (polar)**

| Sub-element | Attribute | Description | Units | Example | Quantity | Default |
|---|---|---|---|---|---|---|
| position | coordinate= “azimuth” | azimuth “theta” of sound location | degrees (−180 ≤ theta ≤ 180) | −22.5 | 1 | |
| position | coordinate= “elevation” | elevation “phi” of sound location | degrees (−90 ≤ phi ≤ 90) | 5.0 | 1 | |
| position | coordinate= “distance” | distance “r” from origin, where 1 is on the unit sphere surface | relative distance value | 0.9 | 0 or 1 | 1.0 |
| width | | horizontal extent | degrees (0 to 360) | 45 | 0 or 1 | 0.0 |
| height | | vertical extent | degrees (0 to 360) | 20 | 0 or 1 | 0.0 |
| depth | | distance extent | Ratio (0 to 1) | 0.2 | 0 or 1 | 0.0 |

For a Cartesian coordinate system, where the position and size values are relative to the cube, where 1 or −1 are on the surface of the unit cube:

TABLE 16

**audioBlockFormat sub-elements for Objects (Cartesian)**

| Sub-element | Attribute | Description | Units | Example | Quantity | Default |
|---|---|---|---|---|---|---|
| position | coordinate="X" | left/right dimension | Relative Units | −0.2 | 1 | |
| position | coordinate="Y" | back/front dimension | Relative Units | 0.1 | 1 | |
| position | coordinate="Z" | bottom/top dimension | Relative Units | −0.5 | 0 or 1 | 0.0 |
| width | | X-width | Relative Units (0 to 1) | 0.03 | 0 or 1 | 0.0 |
| depth | | Y-width | Relative Units (0 to 1) | 0.05 | 0 or 1 | 0.0 |
| height | | Z-width | Relative Units (0 to 1) | 0.07 | 0 or 1 | 0.0 |

The **screenEdgeLock** attribute also exists with the **position** element, which is described in § 5.4.3.1.

The following parameters are independent of the coordinates system used:

TABLE 17

**audioBlockFormat sub-elements for Objects**

| Sub-element | Attribute | Description | Units | Example | Quantity | Default |
|---|---|---|---|---|---|---|
| cartesian | | Specifies coordinate system, if the flag is set to 1 the Cartesian coordinate system is used, otherwise spherical coordinates are used. | 1/0 flag | 1 | 0 or 1 | 0 |
| diffuse | | Describes the diffuseness of an audioObject (if it is diffuse or direct sound) | 0.0 to 1.0 | 0.5 | 0 or 1 | 0 |
| channelLock | maxDistance | If set to 1 a renderer can lock the object to the nearest channel or speaker, rather than normal rendering. The optional maxDistance attribute defines the radius of a sphere around the object's position. If one or more speakers exist in the defined sphere or on its surface, the object snaps to the nearest speaker. If maxDistance is undefined, a default value of infinity is assumed, meaning that the object should snap to the nearest of all speakers (unconditioned channelLock). | 1/0 flag for channelLock,<br><br>float value for maxDistance in the range from 0.0 to 2 (1) sqrt(3) | 1, 1.0 | 0 or 1 | 0 (channel Lock),<br><br>infinity (maxDistance) |
| objectDivergence | azimuthRange (1) | Adjusts the balance between the object's specified position and two other positions specified by the azimuthRange value (symmetrical on both sides of the object at the object's position +/– azimuthRange). A value of 0 for the objectDivergence means no divergence.<br>This attribute shall only be used when the coordinate system is spherical. | 0 to 1.0 for objectDivergence, 0.0 to 180.0 (angle) for azimuthRange | 0.5, 60.0 | 0 or 1 | 0.0, 0.0 |
| | positionRange (1) | Adjusts the balance between the object's specified position and two other positions specified by the positionRange value (symmetrical on both sides of the object at the object's position +/– positionRange along the X-axis).<br>A value of 0 for the objectDivergence means no divergence.<br>This attribute shall only be used when the coordinate system is Cartesian. | 0 to 1.0 for objectDivergence, 0.0 to 1.0 for positionRange | 0.5, 0.25 | 0 or 1 | 0.0, 0.0 |

TABLE 17 (*end*)

| Sub-element | Attribute | Description | Units | Example | Quantity | Default |
|---|---|---|---|---|---|---|
| jumpPosition | | If jumpPosition is set to 1 the position will change instantly from the previous block's position. If set to 0 then interpolation of the position will take the entire length of the block. If the interpolationLength attribute is used, and the jumpPosition value is 1, then the interpolation will take as long as the specified value. The interpolation length should be shorter or equal than the block's duration. | 1/0 flag | 1, 0.05125 | 0 or 1 | 0 |
| | interpolationLength | If the interpolationLength attribute is used, and the jumpPosition value is 1, then the interpolation will take as long as the specified value. The interpolation length should be shorter or equal than the block's duration. | seconds (at least 5 d.p) | 0.05125 | 0 or 1 | Duration of block |
| zoneExclusion ("zone" sub-elements) | | Indicates which speaker/room zones the object should not be rendered through. | see "zone" sub-elements | | 0 or 1 | |
| zone (sub-element of zoneExclusion) | minX<br>maxX<br>minY<br>maxY<br>minZ<br>maxZ | Specifies the corner points of a cuboid in the 3D space that will be excluded from rendering for Cartesian coordinates. Multiple zone elements can be used to specify more complex exclusion shapes. | −1.0 to 1.0 float for each Cartesian attribute.<br>String for a label to describe the exclusion zone | minX=–1.0<br>maxX=1.0<br>minY=–1.0<br>maxY=0.0<br>minZ=–1.0<br>maxZ=1.0<br>"Rear half" | 1.. [1] | |
| | minElevation<br>maxElevation<br>minAzimuth<br>maxAzimuth | Specifies the circular projection onto the sphere for spherical coordinates. Multiple zone elements can be used to specify more complex exclusion shapes. | -180 to 180 float for the spherical azimuth attribute and<br>–90 to 90 float for the spherical elevation attribute.<br>String for a label to describe the exclusion zone | maxElevation=30<br>minElevation=−30<br>minAzimuth=−30<br>maxAzimuth=30<br>"Centre front" | 1.. [1] | |
| screenRef | | Indicates whether the object is screen-related (flag is equal to 1) or not (flag is equal to 0) | 1/0 flag | 0 | 0 or 1 | 0 |

[1] The positionRange or azimuthRange attributes must not be both present in the objectDivergence element.

#### 5.4.3.3.1 Sample code

```
<audioBlockFormat ...>
  <position coordinate="azimuth">-22.5</position>
  <position coordinate="elevation">5.0</position>
  <position coordinate="distance">0.9</position>
  <depth>0.2</depth>
</audioBlockFormat>
```

### 5.4.3.4 If audioChannelFormat.typeDefinition == “HOA”

In scene based audio, a sound scene is represented by a set of coefficient signals. These coefficient signals are the linear weights of spatial orthogonal basis functions (such as spherical or circular harmonics functions). The scene can then be reproduced by rendering these coefficient signals to a target loudspeaker layout or headphones. The program production is decoupled from the reproduction and allows the creation of mixed program material while being agnostic to the number and position of the target loudspeakers. An example of scene-based audio is Higher-Order Ambisonics (HOA).

The definition of audioChannelFormat.typeDefinition == “HOA” is used for scene-based coefficient signals (or components) that use (higher-order) Ambisonics (HOA). Each component can be described by either a combination of degree, order values, and normalization, or an equation.

The HOA components are defined by the degree, order, and normalization values. Degree, order, and normalization are specified in § 11.

If the optional equation sub-element is used, it is recommended that C-style mathematical notation be used for the equation element (e.g. ‘cos(A)*sin(E)’). Its purpose is to allow an informative description of customized or experimental HOA components that cannot be described by the order, degree, and normalization parameters alone.

The normalization, nfcRefDist and screenRef parameters occur in both the audioPackFormat (see § 5.5.5.1) and audioBlockFormat. Therefore, the values of these parameters should be matched in both elements if they are referenced to each other. However, when the parameters are specified within an audioBlockFormat differ from those in the audioPackFormat, the audioBlockFormat values should take priority over those given in the audioPackFormat.

TABLE 18

**audioBlockFormat sub-elements for HOA**

| Sub-element | Description | Units | Example | Quantity | Default | Required |
|---|---|---|---|---|---|---|
| equation | An equation to describe the HOA component | | cos(A)*sin(E) | 0 or 1 | | Optional, used only for descriptive/informative purposes. |
| order | Order of the HOA component | | 1 | 0 or 1 | | Yes |
| degree | Degree of the HOA component | | −1 | 0 or 1 | | Yes |
| normalization | Indicates the normalization scheme of the HOA component (N3D, SN3D, FuMa). | | N3D | 0 or 1 | SN3D | Optional |

TABLE 18 (*end*)

| Sub-element | Description | Units | Example | Quantity | Default | Required |
|---|---|---|---|---|---|---|
| nfcRefDist | Indicates the reference distance of the loudspeaker setup for near-field compensation (NFC). If no nfcRefDist is defined or the value is 0, NFC is not necessary. | metre | 2 | 0 or 1 | 0 | Optional |
| screenRef | Indicates whether the component is screen-related (flag is equal to 1) or not (flag is equal to 0) | 1/0 flag | 0 | 0 or 1 | 0 | Optional |

#### 5.4.3.4.1 Sample code

```
<audioBlockFormat ...>
  <degree>1</degree>
  <order>1</order>
  <normalization>N3D</normalization>
</audioBlockFormat>
```

### 5.4.3.5 If audioChannelFormat.typeDefinition == “Binaural”

This is for binaural representation of audio. Given that binaural consists of two channels, the left and right ear, this is rather simple. As the name of the audioChannelFormat will be either “leftEar” or “rightEar” there is no other metadata required in audioBlockFormat, apart from **gain** and **importance** which are common to all types.

#### 5.4.3.5.1 Sample code

```
<audioBlockFormat .../>
```

## 5.5 audioPackFormat

The audioPackFormat groups together one or more audioChannelFormats that belong together.

Examples of audioPackFormats are ‘stereo’ and ‘5.1’ for channel-based formats. It can also contain references to other packs to allow nesting. The typeDefinition is used to define the type of channels described within the pack. The typeDefinition/typeLabel must match those in the referred audioChannelFormats. The sub-elements within audioPackFormat are dependent upon the typeDefinition or typeLabel of the audioPackFormat element.

### 5.5.1 Attributes

TABLE 19

**audioPackFormat attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| audioPackFormatID | ID for the pack, see § 6 for the use of the audioPackFormatID in typical channel configurations. The yyyy digits of AP_yyyyxxxx_represent the type of audio contained in the pack. | AP_00010001 | Yes |
| audioPackFormatName | Name for the pack | stereo | Yes |
| typeLabel | Descriptor of the type of channel | 0001 | Optional* |
| typeDefinition | Description of the type of channel | DirectSpeakers | Optional* |
| importance | Importance of a pack. Allows a renderer to discard a pack below a certain level of importance. 10 is the most important, 0 is the least. | 10 | Optional |

* At least one of typeLabel or typeDefinition is required.

There are five different defined typeDefinitions:

TABLE 20

**typeDefinitions**

| typeDefinition | typeLabel | Description |
|---|---|---|
| DirectSpeakers | 0001 | For channel-based audio, where each channel feeds a speaker directly |
| Matrix | 0002 | For channel-based audio where channels are matrixed together, such as Mid-Side, Lt/Rt |
| Objects | 0003 | For object-based audio where channels represent audio objects (or parts of objects), so include positional information |
| HOA | 0004 | For scene-based audio where Ambisonics and HOA are used |
| Binaural | 0005 | For binaural audio, where playback is over headphones |
| User Custom | 1yyy to Fyyy | For user custom types. |

### 5.5.2 Sub-elements

TABLE 21

**audioPackFormat sub-elements**

| Sub-element | Description | Example | Quantity |
|---|---|---|---|
| audioChannelFormatIDRef | Reference to an audioChannelFormat | AC_00010001 | 0...* |
| audioPackFormatIDRef | Reference to an audioPackFormat | AP_00010002 | 0...* |
| absoluteDistance | Absolute distance in metres | 4.5 | 0 or 1 |

There is an overall absolute distance parameter, which can be used with the normalized distance parameters specified with the audioBlockFormats, to give absolute distances to each block.

One example of the usage of the absoluteDistance parameter may be to indicate the assumed reference decoding distance (in metres) of a scene-based audio stream. This reference distance may be used in binaural rendering of the rendered soundfield.

If absoluteDistance is negative or undefined, distance based binaural rendering is not intended.

### 5.5.3 Sample code

```
<audioPackFormat audioPackFormatID="AP_000010002" audioPackFormatName="stereo"
typeLabel="0001">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
</audioPackFormat>
```

### 5.5.4 If audioPackFormat.typeDefinition == “Matrix”

If the typeDefinition of the audioPackFormat is set to Matrix, then there are extra sub-elements available to allow the definition of encoding (e.g. Left/Right to Mid/Side), decoding (e.g. Mid/Side to Left/Right) and direct (e.g. Lo/Ro) matrices.

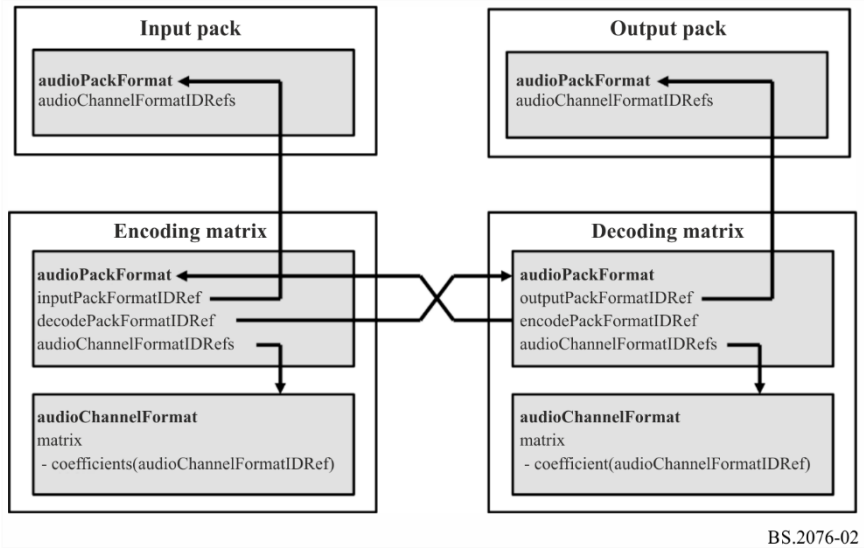
The matrix can either be an encoding, a decoding matrix or a direct matrix. An encoding matrix converts an input audioPackFormat of any type into a matrix-encoded audioPackFormat. A decoding matrix takes matrix-encoded audioPackFormat and converts into a channel-based output audioPackFormat. Related encoding and decoding matrices can be cross-referenced.

“DirectSpeakers” would be the most commonly used for the case of channel-based matrix encoding/decoding and downmixing. For example, Stereo to Mid/Side would be the encoding matrix, and Mid/Side to Stereo would be the decoding matrix.

The diagram in Fig. 2 shows how encoder and decoder matrix audioPackFormats relate to each other, as well in the input and output audioPackFormats and audioChannelFormats.

FIGURE 2

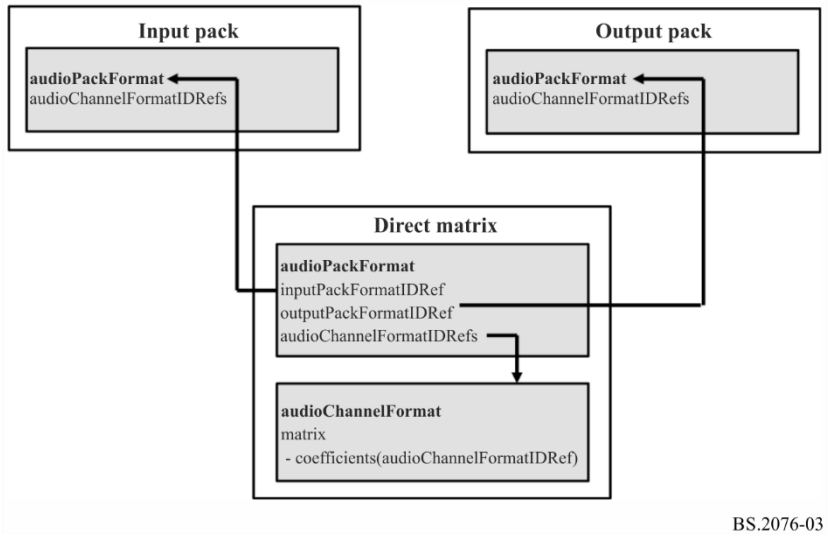
**Encode/Decode Matrix relationships**



The diagram in Fig. 3 shows how a direct matrix audioPackFormat relates to input and output audioPackFormats and audioChannelFormats.

FIGURE 3

**Direct matrix relationships**




### 5.5.4.1 Matrix Sub-elements

The encoding matrix contains an inputPackFormatIDRef, which references a channel-based input pack. It can also contain a list of decodePackFormatIDRefs, which are references to corresponding decoding matrices.

The decoding matrix contains an outputPackFormatIDRef, which reference a channel-based output pack. It can also contain a list of encodePackFormatIDRefs, which are references to corresponding encoding matrices.

The direct matrix contains an inputPackFormatIDRef, which references a channel-based input pack and an outputPackFormatIDRef, which reference a channel-based output pack.

TABLE 22

**audioPackFormat sub-elements for Matrix**

| Sub-element | Description | Example | Quantity |
|---|---|---|---|
| encodePackFormatIDRef | Reference to an encoding matrix audioPackFormat from a decoding matrix. | AP_00020001 | 0…* |
| decodePackFormatIDRef | Reference to a decoding matrix audioPackFormat from an encoding matrix. | AP_00020101 | 0 …* |
| inputPackFormatIDRef | Reference to a channel-based (DirectSpeakers) input audioPackFormat. | AP_00010002 | 0 or 1 |
| outputPackFormatIDRef | Reference to a channel-based (DirectSpeakers) matrix decoded audioPackFormat. | AP_00010002 | 0 or 1 |

#### 5.5.4.2 Sample code

```
<audioPackFormat audioPackFormatID="AP_00021001"
audioPackFormatName="MidSide_Encode" typeLabel="0002"
typeDefinition="Matrix">
  <decodePackFormatIDRef>AP_00021101</decodePackFormatIDRef>
  <inputPackFormatIDRef>AP_00010002</inputPackFormatIDRef>
  <audioChannelFormatIDRef>AC_00021001</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00021002</audioChannelFormatIDRef>
</audioPackFormat>

<audioPackFormat audioPackFormatID="AP_00021101"
audioPackFormatName="MidSide_Decode" typeLabel="0002"
typeDefinition="Matrix">
  <encodePackFormatIDRef>AP_00021001</encodePackFormatIDRef>
  <outputPackFormatIDRef>AP_00010002</outputPackFormatIDRef>
  <audioChannelFormatIDRef>AC_00021101</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00021102</audioChannelFormatIDRef>
</audioPackFormat>
```

### 5.5.5 If audioPackFormat.typeDefinition == “HOA”

If the audioPackFormat element is of HOA type then the following sub-elements can be defined. These parameters provide defaults for the audioBlockFormat parameters corresponding to the HOA-type audioChannelFormat definitions within this audioPackFormat. The normalization, nfcRefDist and screenRef parameters occur in both the audioPackFormat and audioBlockFormat (see § 5.4.3.4). Therefore, the values of these parameters should be matched in both elements if they are referenced to each other. However, when the parameters are specified within an audioBlockFormat differ from those in the audioPackFormat, the audioBlockFormat values take priority over those given in the audioPackFormat.

#### 5.5.5.1 HOA Sub-elements

TABLE 23

**audioPackFormat sub-elements for HOA**

| Sub-element | Description | Units | Example | Quantity | Default | Required |
|---|---|---|---|---|---|---|
| normalization | Indicates the normalization scheme of the HOA content (N3D, SN3D, FuMa). | | N3D | 0 or 1 | SN3D | Optional |
| nfcRefDist | Indicates the reference distance of the loudspeaker setup for near-field compensation (NFC). If no nfcRefDist is defined or the value is 0, NFC is not necessary. | metre | 2 | 0 or 1 | 0 | Optional |
| screenRef | Indicates whether the content is screen-related (flag is equal to 1) or not (flag is equal to 0) | 1/0 flag | 0 | 0 or 1 | 0 | Optional |

## 5.6 audioObject

An audioObject establishes the relationship between the content, the format via audio packs, and the assets using the track UIDs. AudioObjects can be nested and so they can refer to other audioObjects.

### 5.6.1 Attributes

TABLE 24

**audioObject attributes**

| Attribute | Description | Example | Required | Default |
|---|---|---|---|---|
| audioObjectID | ID of the object | AO_1001 | Yes | - |
| audioObjectName | Name of the object | dialogue_stereo | Yes | - |
| start | Start time for the object, relative to the start of the audioProgramme. The start time is in the time format as described in § 5.11. | 00:00:00.00000 or 00:00:00.00000S48000 | Optional | 00:00:00.00000 |
| duration | Duration of object. The duration is in the time format as described in § 5.11. | 00:02:00.00000 or 00:02:00.00000S48000 | Optional | duration of audioProgramme |
| dialogue | If the audio is not dialogue set a value of 0; if it contains only dialogue a value of 1; if it contains both then a value of 2. | 0 | Optional | 2 |
| importance | Importance of an object. Allows a renderer to discard an object below a certain level of importance. 10 is most important, 0 least. | 10 | Optional | 10 |
| interact | Set to 1 if a user can interact with the object, 0 if not. | 1 | Optional | 0 |
| disableDucking | Set to 1 to disallow automatic ducking of object, 0 to allow ducking | 0 | Optional | 0 |

### 5.6.2 Sub-elements

TABLE 25

**audioObject sub-elements**

| Sub-element | Attribute | Description | Units/Type | Example | Quantity |
|---|---|---|---|---|---|
| audioPackFormatIDRef | | Reference to an audioPackFormat for format description | ID String | AP_00010001 | 0…* |
| audioObjectIDRef | | Reference to another audioObject | ID String | AO_1002 | 0…* |
| audioObjectLabel | language | Definition of audioObject label. The language attribute can be used for definition of multiple audioObject labels in different languages. See Table 26. | String | “Dialogue” language=”en” | 0…* |

TABLE 25 (*end*)

| Sub-element | Attribute | Description | Units/Type | Example | Quantity |
|---|---|---|---|---|---|
| audioComplementary ObjectGroupLabel | language | Definition of a label for a group of complementary audioObjects. The language attribute can be used for definition of multiple audioComplementaryObjectGroup labels in different languages. See Table 27. | String | “主音声” language=“jp” | 0…* |
| audioComplementary ObjectIDRef | | Reference to another audioObject that is complementary to the object, e.g. to describe mutually exclusive languages. | ID String | AO_1003 | 0…* |
| audioTrackUIDRef | | Reference to an audioTrackUID (when using a BW64 file according to [7] this is listed in the <*chna*> chunk) | ID String | ATU_00000001 | 0…* |
| audioObjectInteraction | | Specification of possible user interaction with the object. | - | - | 0 or 1 |
| gain | gainUnit | Definition of a gain value to be applied to all audio samples referenced by the audioObject. The default value is 1.0. An optional gainUnit attribute (either ‘linear’ or ‘dB’) can be used to define the unit of the gain value. The default unit is ‘linear’. For a detailed description of the application of this gain value see § 12. | Linear or logarithmic gain value | 0.5 (linear), −6.0 (dB) | 0 or 1 |
| headLocked | | Indicates if the perceived location of the audio element is locked to the head (flag = 1) or not locked (flag = 0)<br>See § 9.3<br>Default Value is 0 | 0/1 flag | 1 | 0 or 1 |
| positionOffset *(when polar coordinates are used* | coordinate= “azimuth” | Apply an offset to the “azimuth” angle to all elements in the audioObjects. | Degrees | 30.0 | 0 or 1 |
| | coordinate= “elevation” | Apply an offset to the “elevation” angle to all elements in the audioObjects. | Degrees | 15.0 | 0 or 1 |
| | coordinate= “distance” | Apply an offset of “distance” to all elements in the audioObjects. | Normalised distance | 0.9 | 0 or 1 |
| positionOffset *(when Cartesian coordinates are used)* | coordinate= “X” | Apply an offset of “X” axis to all elements in the audioObjects. | Normalised value | −0.2 | 0 or 1 |
| | coordinate= “Y” | Apply an offset of “Y” axis to all elements in the audioObjects. | Normalised value | 0.1 | 0 or 1 |
| | coordinate= “Z” | Apply an offset of “Z” axis to all elements in the audioObjects. | Normalised value | −0.5 | 0 or 1 |
| mute | | Status of the audioObject to play back or not. Set to 0 if the object is played back (default). Set to 1 if the object is muted. | | 1 | 0 or 1 |
| alternativeValueSet | alternativeValue SetID | An alternative set of parameters that will be used if the alternativeValueSetID is referenced by an audioProgramme or audioContent element.<br>See § 5.6.5 for sub-elements. | | | 0…* |

If the value of audioTrackUIDRef is set to ATU_00000000 then it does not refer to a track in the file, but refers to a silent or empty track. This could useful for multichannel formats where some of the channels are not being used, so instead of storing zero value samples in the file, this silent track is used instead thus saving space in the file.

TABLE 26

**audioObjectLabel attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| language | The language attribute can be used for definition of multiple audioObject labels in different languages. The language code is given as a 2- or 3-character code as specified by ISO 639-1 or ISO 639-2. Both ISO 639-2/B and ISO 639-2/T may be used. | eng | No |

The audioComplementaryObjectGroupLabel element contains a textual label for a set of mutually exclusive audioObjects, e.g. language tracks that contain the same dialogue in different dub versions.

The audioComplementaryObjectGroupLabel element should only be included in one corresponding parent audioObject for each set of mutually exclusive contents. The same parent audioObject should be used that also contains the audioComplementaryObjectIDRef sub-elements.

TABLE 27

**audioComplementaryObjectGroupLabel attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| language | Attribute defining the language of the parent audioComplementaryObjectGroupLabel. The language code is given as a 2- or 3-character code as specified by ISO 639-1 or ISO 639-2. Both ISO 639-2/B and ISO 639-2/T may be used. | eng | No |

### 5.6.3 audioComplementaryObjectIDRef

The audioComplementaryObjectIDRef element contains a reference to another audioObject that is complementary to the parent audioObject. A list of audioComplementaryObjectIDRefs can therefore be used to describe mutually exclusive content, e.g. language tracks that contain the same dialogue in different dub versions (“XOR” relationship).

To avoid cross-references between audioComplementaryObjectIDRefs of several audioObjects, the audioComplementaryObjectIDRef sub-element should only be included in one corresponding parent audioObject for each set of mutually exclusive contents. The parent audioObject with the audioComplementaryObjectIDRefs should be the one that contains the default version of the set of mutually exclusive contents.

### 5.6.4 audioObjectInteraction sub-element

An audioObjectInteraction element describes any possible user interaction with the corresponding parent audioObject. It should be present only if the “Interact” attribute of the parent audioObject is set to 1. In case the “Interact” attribute of the parent audioObject is set to 0, any audioObjectInteraction element should be ignored. The audioObjectInteraction element has the following attributes and sub-elements.

TABLE 28

**audioObjectInteraction attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| onOffInteract | Set to 1 if a user can switch the object on or off, 0 if not. | 1 | Yes |
| gainInteract | Set to 1 if a user can change the gain of the object, 0 if not. | 1 | Optional |
| positionInteract | Set to 1 if a user can change the position of the object, 0 if not. | 0 | Optional |

If the onOffInteract attribute is set to 1, the audioObject can be switched on or off by the user. In the case that the "gainInteract" attribute is set to 1, the gain of the audioObject can be changed by a user in accordance with the following "gainInteractionRange" element. In the case that the "positionInteract" attribute is set to 1, the positions of the audioBlockFormats in the parent audioObject can be changed by a user in accordance with the following "positionInteractionRange" element.

TABLE 29

**audioObjectInteraction sub-elements**

| Sub-element | Attribute | Bound attribute | Description | Units | Example |
|---|---|---|---|---|---|
| gainInteractionRange | N/A | min | Minimum linear gain factor or logarithmic gain offset of a possible user gain interaction. For a detailed description of the application of this gain-related parameter see § 12. (NOTE: The earlier versions of this Recommendation contained the following formulae to describe the application of the minimum bound of the gain interaction range "Linear gain: gainMin = gain (or 1.0 if not defined) * gainInteractionRangeMin". This formula was not correct as it was undefined to what ADM element or parameter "gain" in the formula was referring. In the current version (Rec. ITU-R BS.2076-2), this erroneous formula was removed. The gainInteractionRange boundaries should be interpreted as described in § 12.) | Linear or logarithmic (dB) gain value | 0.5 |
| | N/A | max | Maximum linear gain factor or logarithmic gain offset of possible user gain interaction. For a detailed description of the application of this gain-related parameter see § 12. (NOTE: The earlier versions of this Recommedation contained the following formulae to describe the application of the maximum bound of the gain interaction range (Linear gain: gainMax = gain (or 1.0 if not defined) * gainInteractionRangeMax". This formula was not correct as it was undefined to what ADM element or parameter "gain" was referring in the formula. In the current version (Rec. ITU-R BS.2076-2), this erroneous formula was removed. The gainInteractionRange boundaries should be interpreted as described in § 12. | Linear of logarithmic (dB) gain value | 1.2 |
| | gainUnit | | Unit for attribute of 'gain'. If gainUnit is not used, 'linear' unit is used. | | linear / dB |

TABLE 29 (*end*)

| Sub-element | Attribute | Bound attribute | Description | Units | Example |
|---|---|---|---|---|---|
| positionInteractionRange *(when polar coordinates are used)* | coordinate= “azimuth” | min | Minimum azimuth offset value of possible user position interaction | Degrees | –30.0 |
| | coordinate= “azimuth” | max | Maximum azimuth offset value of possible user position interaction | Degrees | +30.0 |
| | coordinate= “elevation” | min | Minimum elevation offset value of possible user position interaction | Degrees | –15.0 |
| | coordinate= “elevation” | max | Maximum elevation offset value of possible user position interaction | Degrees | +15.0 |
| | coordinate= “distance” | min | Minimum normalized distance of possible user position interaction | 0 to 1 | 0.5 |
| | coordinate= “distance” | max | Maximum normalized distance of possible user position interaction | 0 to 1 | 0.5 |
| positionInteractionRange *(when Cartesian coordinates are used)* | coordinate= “X” | min | Minimum X-axis offset value of possible user position interaction | Normalized Units | –0.5 |
| | coordinate= “X” | max | Maximum X-axis offset value of possible user position interaction | Normalized Units | +0.5 |
| | coordinate= “Y” | min | Minimum Y-axis offset value of possible user position interaction | Normalized Units | –0.2 |
| | coordinate= “Y” | max | Maximum Y-axis offset value of possible user position interaction | Normalized Units | 0.0 |
| | coordinate= “Z” | min | Minimum Z-axis offset value of possible user position interaction | Normalized Units | 0.1 |
| | coordinate= “Z” | max | Maximum Z-axis offset value of possible user position interaction | Normalized Units | 0.4 |

### 5.6.4.1 Sample code

```
<audioObjectInteraction onOffInteract="1" gainInteract="1" positionInteract="1">
  <positionInteractionRange coordinate="elevation" bound="min">
     -10.0
  </positionInteractionRange>
  <positionInteractionRange coordinate="elevation" bound="max">
     +10.0
  </positionInteractionRange>
  <positionInteractionRange coordinate="azimuth" bound="min">
     -30.0
  </positionInteractionRange>
  <positionInteractionRange coordinate="azimuth" bound="max">
     +30.0
  </positionInteractionRange>
</audioObjectInteraction>
```

If an *audioObject* allows interaction, the result of a user-imposed change to an attribute that can be set by the user should be within the limits of the interaction range of that *audioObject*. In this context, a “change” is the difference between a condition before and after the interaction.

The resultant overall playback gain of a sound source is the combination of the attributes of the gain sub-elements of the *audioBlockFormat* and all changes caused by interaction in the hierarchy of *audioObjects* that refer to the *audioBlockFormat* (see § 12).

### 5.6.5 alternativeValueSet sub-element

The alternativeValueSet sub-element allows an alternative set of parameters for the audioObject to be defined. The parameters defined in this sub-element will take precedence over the same parameters in the parent audioObject element. The parameters defined in the parent audioObject that have not been defined in the alternativeValueSet shall be used in that alternativeValueSet. Multiple alternativeValueSets can be defined in an audioObject to allow multiple variations to be defined. Table 30 lists the sub-elements that are contained within alternativeValueSet, and each has the same specification of the same sub-elements in the parent audioObject as listed in Table 25.

TABLE 30

**alternativeValueSet sub-elements**

| Sub-element | Note |
|---|---|
| audioObjectLabel | See Table 25 for attributes, descriptions, examples, units and quantities. |
| audioObjectInteraction | |
| gain | |
| headLocked | |
| positionOffset | |
| mute | |

#### 5.6.5.1 alternativeValueSetID attribute

The alternativeValueSet sub-element shall use an alternativeValueSetID attribute and the ID shall be in this format: AVS_wwww_zzzz, where ‘w’ and ‘z’ are hexadecimal digits. The ‘wwww’ shall match the ‘wwww’ of the parent audioObjectID, and the ‘zzzz’ shall be a unique value for each alternativeValueSet sub-element used within the parent audioObject.

The alternativeValueSetID may be referenced from either audioProgramme or audioContent.

#### 5.6.5.2 Sample code

```
<audioObject audioObjectID="AO_1001" audioObjectName="Effects">
  <gain>1.0</gain>
  <alternativeValueSet alternativeValueSetID=”AVS_1001_0001”>
    <gain>1.5</gain>
  </alternativeValueSet>
  <alternativeValueSet alternativeValueSetID=”AVS_1001_0002”>
    <gain>0.5</gain>
  </alternativeValueSet>
  …
</audioObject>
```

### 5.6.6 Sample code

```
<audioObject audioObjectID="AO_1001" audioObjectName="Dialogue_stereo">
  <audioPackFormatIDRef>AP_00010001</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
</audioObject>
```

### 5.6.7 Nested audioObjects and timing parameters

When audioObject elements are nested the start time of the audioObject is still relative to the start of the programme, not the relative to the audioObject that refers to it. It is required to ensure that any audioObject that is referred from another audioObject does not have a start time earlier than the referent, nor does it have an end time (i.e. start + duration) after the referent.

An audioObject element should not reference itself, nor can a loop of references be used (e.g. AO_1001 -> AO_1002 -> AO_1003 -> AO_1001 would be a loop and therefore illegal).

## 5.7 audioContent

An audioContent element describes the content of one component of a programme (e.g. background music), and refers to audioObjects to tie the content to its format. This element includes loudness metadata.

### 5.7.1 Attributes

TABLE 31

**audioContent attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| audioContentID | ID of the content | ACO_1001 | Yes |
| audioContentName | Name of the content | Music | Yes |
| audioContentLanguage | Language of the content (as a String). It is recommended to use a language code to identify the language. The language code can be given as a 2- or 3-character code as specified by ISO 639-1 or ISO 639-2. Both ISO 639-2/B and ISO 639-2/T may be used. | en | Optional |

### 5.7.2 Sub-elements

TABLE 32

**audioContent sub-elements**

| Sub-element | Attribute | Description | Example | Quantity |
|---|---|---|---|---|
| audioContentLabel | language | Definition of an audioContent label (as a String). The language attribute can be used for definition of multiple audioContent labels in different languages. It is recommended to use a language code to identify the language. The language code should be given as a 2- or 3-character code as specified by ISO 639-1 or ISO 639-2. Both ISO 639-2/B and ISO 639-2/T may be used. | “News” language=”en” | 0…* |
| audioObjectIDRef | | Reference to audioObject | AO_1001 | 1…* |
| loudnessMetadata | | See § 5.7.4 | | 0 … * |

TABLE 32 (*end*)

| Sub-element | Attribute | Description | Example | Quantity |
|---|---|---|---|---|
| dialogue | | If the audio is not dialogue set a value of 0; if it contains only dialogue set a value of 1; if it contains both then set a value of 2. | 0 | 0 or 1 |
| alternativeValueSetIDRef | | Reference to an alternativeValueSet within an audioObject. | AVS_1001_0001 | 0…* |

As it is possible to include multiple alternativeValueSetIDRef sub-elements within an audioContent element, it should be ensured that the alternativeValueSetIDRef only references one alternativeValueSet within the same audioObject. This should be done by inspecting the alternativeValueSet ID digits. The ID has the format: AVS_wwww_zzzz, where wwww matches the digits in the audioObject ID. Therefore, to ensure an audioObject is not referenced multiple times, each alternativeValueSetIDRef in an audioContent shall have unique wwww digits.

### 5.7.3 dialogue

This optional element specifies the kind of content that is included in the parent audioContent. The Dialogue sub-element can take the values 0 (no dialogue), 1 (pure dialogue) or 2 (mixed). It has an attribute that specifies the type of content using defined lists (enumerators) of content kinds.

The attribute is dependent on the value of the Dialogue element.

TABLE 33

**dialogue attributes**

| Value of dialogue | Attribute | Description | Example |
|---|---|---|---|
| 0 | nonDialogueContentKind | ID of the contained content kind (enumerator, see specification below) | 0 |
| 1 | dialogueContentKind | ID of the contained content kind (enumerator, see specification below) | 0 |
| 2 | mixedContentKind | ID of the contained content kind (enumerator, see specification below) | 0 |

TABLE 34

**dialogue types**

| nonDialogueContentKind | Description |
|---|---|
| 0 | Undefined |
| 1 | Music |
| 2 | Effect |
| **dialogueContentKind** | **Description** |
| 0 | Undefined |
| 1 | (Storyline) dialogue |
| 2 | Voiceover |
| 3 | Spoken subtitle |

TABLE 34 (*end*)

| nonDialogueContentKind | Description |
|---|---|
| 4 | Audio description/visually impaired |
| 5 | Commentary |
| 6 | Emergency |
| **mixedContentKind** | **Description** |
| 0 | Undefined |
| 1 | Complete main |
| 2 | Mixed |
| 3 | Hearing impaired |

### 5.7.4 loudnessMetadata attributes and sub-elements

TABLE 35

**loudnessMetadata attributes**

| Attribute | Description | Example |
|---|---|---|
| loudnessMethod | The method or algorithm used to calculate the loudness. | “ITU-R BS.1770” |
| loudnessRecType | The loudnessRecType indicates which regional recommended practice was followed in the loudness correction of the audio | “EBU R128” |
| loudnessCorrectionType | The correction type is used to indicate what correction the audio, for example, file-based or real-time. | “File-based” |

The audio could be measured by various means, relating to loudness algorithm, regional recommended practice followed, and by what correction type. The loudnessMethod or algorithm used will typically be BS.1770, but in the future, there could be newer methods. The loudnessRecType indicates the regional recommended practice that was followed as a character string, such as “EBU R128”, “ATSC A/85”, “ARIB TR B32” or “FreeTV OP59”. The loudnessCorrectionType specifies how the audio has been correlated: in an off-line file-based or a real-time process.

TABLE 36

**loudnessMetadata sub-elements**

| Sub-element | Description | Units | Example |
|---|---|---|---|
| integratedLoudness | Integrated loudness value | LKFS/LUFS | −23.0 |
| loudnessRange | Loudness range | LU | 10.0 |
| maxTruePeak | Maximum true-peak | dBTP | −2.3 |
| maxMomentary | Maximum momentary loudness | LKFS/LUFS | −19.0 |
| maxShortTerm | Maximum short- term loudness | LKFS/LUFS | −21.2 |
| dialogueLoudness | Loudness of the average dialogue | LKFS/LUFS | −24.0 |

NOTE – ITU-R BS.1770 uses LKFS for loudness units, and the EBU uses LUFS. Both units are identical, and the model does not require the units to be expressed in the metadata.

### 5.7.5 Sample code

```
<audioContent audioContentID="ACO_1001" audioContentName="Music">
  <audioContentLabel language="eng">Music</audioContentLabel>
  <audioContentLabel language="deu">Musik</audioContentLabel>
  <audioObjectIDRef>AO_1001</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-23.0</integratedLoudness>
    <maxTruePeak>-2.3</maxTruePeak>
  </loudnessMetadata>
</audioContent>
```

## 5.8 audioProgramme

An audioProgramme element refers to a set of one or more audioContents that are combined to create a full audio programme. It contains start and end times for the programme, which can be used for alignment with video times. Loudness metadata is also included to allow the programme's loudness to be recorded.

When more than one audioProgramme is included in a file, and there is no other information to decide which one to choose for playback, then the default audioProgramme is the one with the lowest ID value.

### 5.8.1 Attributes

TABLE 37

**audioProgramme attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| audioProgrammeID | ID of the programme | APR_1001 | Yes |
| audioProgrammeName | Name of the programme | | Yes |
| audioProgrammeLanguage | Language of the dialogue content contained in this programme (as a String). It is recommended to use a language code to identify the language. The language code can be given as a 2- or 3-character code as specified by ISO 639-1 or ISO 639-2. Both ISO 639-2/B and ISO 639-2/T may be used. | fr | Optional |
| start | Start time for the programme. The start time is in the time format as described in § 5.11. | 00:00:10.00000 or 00:00:10.00000S48000 | Optional |
| end | End time for the programme. The end time is in the time format as described in § 5.11. | 00:10:00.00000 or 00:10:00.00000S48000 | Optional |
| maxDuckingDepth | Indicates the maximum amount of automatic ducking allowed for every audioObject in the programme. Range is 0 to −62 dB | | Optional |

### 5.8.2 Sub-elements

TABLE 38

**audioProgramme sub-elements**

| Sub-element | Attribute | Description | Example | Quantity |
|---|---|---|---|---|
| audioProgrammeLabel | language | Definition of audioProgramme label. The language attribute can be used for definition of multiple audioProgramme labels in different languages. The language code should be given as a 2- or 3-character code as specified by ISO 639-1 or ISO 639-2. Both ISO 639-2/B and ISO 639-2/T may be used | "Venue" language="en" | 0…* |
| audioContentIDRef | | Reference to content | ACO_1001 | 1…* |
| loudnessMetadata | - | See § 5.8.4 | | 0 … * |
| audioProgrammeReferenceScreen | - | Specification of a reference/ production/monitoring screen size for the audioProgramme, see § 5.8.3. If the reference screen-size is not given, a default screen-size is implicitly defined (see § 10.6). | | 0 or 1 |
| authoringInformation | | See § 5.8.6 | | 0 or 1 |
| alternativeValueSetIDRef | | Reference to an alternativeValueSet within an audioObject. | AVS_1001_0001 | 0…* |

As it is possible to include multiple alternativeValueSetIDRef sub-elements within an audioProgramme element, it should be ensured that the alternativeValueSetIDRef only references one alternativeValueSet within the same audioObject. This should be done by inspecting the alternativeValueSet ID digits. The ID has the format: AVS_wwww_zzzz, where wwww matches the digits in the audioObject ID. Therefore, to ensure an audioObject is not referenced multiple times, each alternativeValueSetIDRef in an audioProgramme shall have unique wwww digits.

### 5.8.3 audioProgrammeReferenceScreen

An audioProgrammeReferenceScreen element describes a reference/production/monitoring screen that was used by the content creator during the production of the content of this audioObject. The screen can be described using either polar coordinates or Cartesian coordinates, but not both (see Fig. 4).

TABLE 39

**audioProgrammeReferenceScreen attributes**

| Attribute | Description | Example |
|---|---|---|
| aspectRatio | Aspect ratio of the screen (proportional relationship between its width and its height (with respect to the image dimensions)) | 1.78, 1.6 |

For when polar coordinates are used:

TABLE 40A

**audioProgrammeReferenceScreen sub-elements**

| Sub-element | Coordinate Attribute | Description | Units | Example |
|---|---|---|---|---|
| screenCentrePosition | azimuth | Azimuth angle of the centre of the screen | degrees | +30.0 |
| | elevation | Elevation angle of the centre of the screen | degrees | −15.0 |
| | distance | Normalized distance to the centre of the screen. Default is 1.0 | Normalized units (0.0 to 1.0) | 1.0 |
| screenWidth | azimuth | Width of the screen in polar coordinates (azimuth opening angle theta) | degrees ($0 < \text{theta} \leq 180$) | +58.0 or +96.0 |

For when Cartesian coordinates are used:

TABLE 40B

| Sub-element | Coordinate Attribute | Description | Units | Example |
|---|---|---|---|---|
| screenCentrePosition | X | X-coordinate of the centre of the screen | Normalized units ($\text{abs}(X) \leq 1$) | −0.3 |
| | Y | Y-coordinate of the centre of the screen | Normalized units ($\text{abs}(Y) \leq 1$) | −0.2 |
| | Z | Z-coordinate of the centre of the screen | Normalized units ($\text{abs}(Z) \leq 1$) | 1.0 |
| screenWidth | X | Width of the screen in Cartesian coordinates (width of the screen on the X-axis) | $0 < X \leq 2$ | 0.8 |

### 5.8.4 loudnessMetadata attributes and sub-elements

TABLE 41

**loudnessMetadata attributes**

| Attribute | Description | Example |
|---|---|---|
| loudnessMethod | The method or algorithm used to calculate the loudness. | “ITU-R BS.1770” |
| loudnessRecType | The loudnessRecType indicates which regional recommended practice was followed in the loudness correction of the audio | “EBU R128” |
| loudnessCorrectionType | The correction type is used to indicate what correction the audio, for example, file-based or real-time. | “File-based” |

The audio could be corrected or normalized by numerous means, relating to loudness algorithm, regional recommended practice followed, and by what correction type. The loudnessMethod or algorithm used will typically be “ITU-R BS.1770” as defined in Recommendation ITU-R BS.1770 [5], but in the future, there could be newer methods. The loudnessRecType indicates the regional recommended practice that was followed as a character string, such as “EBU R128”, “ATSC A/85”, “ARIB TR B32” or “FreeTV OP59”. The loudnessCorrectionType specifies how the audio has been correlated: in an off-line file-based or a real-time process.

TABLE 42

**loudnessMetadata sub-elements**

| Sub-element | Description | Units | Example |
|---|---|---|---|
| integratedLoudness | Integrated loudness value | LKFS/LUFS | −23.0 |
| loudnessRange | Loudness range | LU | 10.0 |
| maxTruePeak | Maximum true-peak | dBTP | −2.3 |
| maxMomentary | Maximum momentary loudness | LKFS/LUFS | −19.0 |
| maxShortTerm | Maximum short- term loudness | LKFS/LUFS | −21.2 |
| dialogueLoudness | Loudness of the average dialogue | LKFS/LUFS | −24.0 |

NOTE – ITU-R BS.1770 uses LKFS for loudness units, and the EBU uses LUFS. Both units are identical, and the model does not require the units to be expressed in the metadata.

### 5.8.5 Sample code

```
<audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Documentary">
  <audioProgrammeLabel language="eng">Default Mix</audioProgrammeLabel>
  <audioProgrammeLabel language="deu">Standard Mix</audioProgrammeLabel>
  <audioContentIDRef>ACO_1001</audioContentIDRef>
  <audioContentIDRef>ACO_1002</audioContentIDRef>
</audioProgramme>
```

### 5.8.6 authoringInformation

TABLE 43

**authoringInformation sub-elements**

| Sub-element | Description | Quantity |
|---|---|---|
| referenceLayout | The reference layout describes the loudspeaker layout for which the content of the audioProgramme was originally produced for. In that sense it represents the optimal loudspeaker layouts from the content creator's point of view. See Table 44. | 0 …* |
| renderer | See Table 45 and Table 46. | 0 … * |

TABLE 44

**referenceLayout sub-elements**

| Sub-element | Description | Example | Quantity |
|---|---|---|---|
| audioPackFormatIDRef | Reference to an audioPackFormat used as the reference layout during production. The referenced layout can either be part of the Common Definitions in Recommendation ITU-R BS.2094 or contained in the local ADM code itself. In case a reproduction technique is used during production that makes use of a virtual loudspeaker setup (e.g. binaural rendering or soundbar rendering), the referenceLayout should reference the virtual loudspeaker layout. | AP_00010003 | 1 |

TABLE 45

**renderer attributes**

| Attributes | Description | Example | Required |
|---|---|---|---|
| uri | Renderer uri used in production and monitoring. | urn:itu:bs:2127:0:itu_adm_renderer | Yes |
| name | Renderer name used in production and monitoring. | Rec, ITU-R BS.2127 | Optional |
| version | Version number of the renderer. | “1.0.0” | Optional |

TABLE 46

**renderer sub-elements**

| Sub-element | Description | Example | Quantity |
|---|---|---|---|
| audioPackFormatIDRef | Reference to an audioPackFormat used in production and monitoring. | AP_00010003 | 1…* |

### 5.8.7 Sample code

```
<audioFormatExtended version= "ITU-R_BS.2076-2">
  <audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="MyProgramme">
    <authoringInformation>
      <renderer uri="urn:itu:bs:2127:0:itu_adm_renderer">
       <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
       <audioPackFormatIDRef>AP_00010017</audioPackFormatIDRef>
      </renderer>
    </authoringInformation>
  </audioProgramme>
</audioFormatExtended>
```

## 5.9 audioTrackUID

The audioTrackUID uniquely identifies a track or asset within a file or recording of an audio scene. This element contains information about the bit-depth and sample-rate of the track. For PCM audio, the audioStreamFormat and the audioTrackFormat may be omitted. Then the audioTrackUID has to refer to the corresponding audioChannelFormat and the same number is used for the 'yyyyxxxx' parts of AT_yyyyxxxx_zz, AS_yyyyxxxx and AC_yyyyxxxx. It also contains sub-elements that allow the model to be used for non-BW64 applications by performing the job of the <*chna*> chunk. When using the model with MXF files the audioMXFLookUp sub-element (which contains sub-elements to refer to the audio essences in the file) is used.

### 5.9.1 Attributes

TABLE 47

**audioTrackUID attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| UID | The actual UID value | ATU_00000001 | Yes |
| sampleRate | Sample rate of track in Hz | 48000 | Optional |
| bitDepth | Bit-depth of track in bits | 24 | Optional |

### 5.9.2 Sub-elements

TABLE 48

**audioTrackUID sub-elements**

| Sub-element | Description | Example | Quantity |
|---|---|---|---|
| audioMXFLookUp | See § 5.9.3 | | 0 or 1 |
| audioTrackFormatIDRef | Reference to an audioTrackFormat description | AT_00010001_01 | 0 or 1 |
| audioChannelFormatIDRef | Reference to an audioChannelFormat description. This element is used only if an audioTrackFormat is omitted for PCM audio. Then the 'yyyyxxxx' parts of AC_yyyyxxxx and AT_yyyyxxxx_zz are the same number. | AC_00010001 | 0 or 1 |
| audioPackFormatIDRef | Reference to an audioPackFormat description | AP_00010002 | 0 or 1 |

### 5.9.3 MXF sub-elements

MXF has different meanings for the terms ‘track’ and ‘channel’ from their use in the ADM. In MXF ‘track’ is the storage medium containing audio or video, and for audio this ‘track’ can be sub-divided into ‘channels’.

TABLE 49

**MXF sub-elements**

| Sub-element | Description | Type | Example |
|---|---|---|---|
| packageUIDRef | Reference to an MXF package | UMID string | urn:smpte:umid: 060a2b34.01010105.01010f20.13000000. 540bca53.41434f05.8ce5f4e3.5b72c985 |
| trackIDRef | Reference to an MXF track | int | MXFTRACK_3 |
| channelIDRef | Reference to a channel track | int | MXFCHAN_1 |

### 5.9.4 Sample code

```
<audioTrackUID UID="ATU_00000001" sampleRate="48000" bitDepth="24"/>
```

## 5.10 audioFormatExtended

AudioFormatExtended is the parent element, containing all the ADM elements.

### 5.10.1 Sub-elements

TABLE 50

**audioFormatExtended sub-elements**

| Sub-element | Description | Quantity |
|---|---|---|
| audioProgramme | Description of the whole audio programme. | 0…* |
| audioContent | Description of the content of some audio within the programme. | 0…* |
| audioObject | The link between the actual audio tracks and their format. | 0…* |
| audioPackFormat | A description of a pack of channels that relate together. | 0…* |
| audioChannelFormat | A description of an audio channel. | 0…* |
| audioStreamFormat | A description of an audio stream. | 0…* |
| audioTrackFormat | A description of an audio track. | 0…* |
| audioTrackUID | The unique identifier for an actual audio track. | 0…* |

None of elements in Table 50 are mandatory within an ADM file. For example, a file that only consists of Common Definition tracks would not contain any audioTrackFormat, audioStreamFormat, audioChannelFormat and audioPackFormat elements. While it is preferable for ADM files to contain at least one audioProgramme and audioContent element, it is still valid for them to be omitted (for example in temporary or test files).

### 5.10.2 Attributes

TABLE 51

**audioFormatExtended attributes**

| Attribute | Description | Example | Required |
|---|---|---|---|
| version | ADM Recommendation name and revision number | “ITU-R_BS.2076-2” | Yes |

The version name is used to indicate which version of the ADM is used. If the version attribute is missing then the ADM is assumed to be Recommendation ITU-R BS.2076-0, as this version of the ADM did not contain this version attribute. For any later version of the ADM, then the version attribute should be included with the relevant name.

The version name for this particular update of the Recommendation is “ITU-R_BS.2076-2”.

### 5.10.3 Sample code

```
<audioFormatExtended version=”ITU-R_BS.2076-2”>
  ...
</audioFormatExtended>
```

## 5.11 Time parameters format

Time-related parameters shall have the format of ‘hh:mm:ss.zzzzz’ or ‘hh:mm:ss.zzzzzSfffff’.

‘hh:mm:ss.zzzzz’ indicates hours, minutes, seconds. The number of decimal places for the seconds should be a minimum of 5. There should be enough decimal places used for sample-accurate timing. For example, 01:34:16.25000.

‘hh:mm:ss.zzzzzSfffff’ indicates hours, minutes, seconds with a fractional representation of sub-seconds. The zzzzz digits represent the numerator of a fraction, and the fffff digits represent the denominator. The number of digits for zzzzz and fffff should be at least five each. This format allows a sample-based representation of time to be used, where zzzzz is the number of samples, and fffff is the sample-rate. The value of zzzzz should be less than fffff to ensure a fraction less than one. Both values should not be negative, and fffff should be greater than zero. For example, 01:34:16.12000S48000 is the same as 01:34:16.25000.

# 6 Use of IDs

The ID attributes in each of the elements have three main purposes: to allow the elements to reference each other, to provide a unique identification for each defined element, and to provide a logical numerical representation of the contents of the element. The ID for each element follows the following format:

TABLE 52

**Element ID formats**

| Element | ID format |
|---|---|
| audioPackFormat | AP_yyyyxxxx |
| audioChannelFormat | AC_yyyyxxxx |
| audioBlockFormat | AB_yyyyxxxx_zzzzzzzz |
| audioStreamFormat | AS_yyyyxxxx |
| audioTrackFormat | AT_yyyyxxxx_zz |
| audioProgramme | APR_wwww |

TABLE 52 (*end*)

| Element | ID format |
|---|---|
| audioContent | ACO_wwww |
| audioObject | AO_wwww |
| alternativeValueSet | AVS_wwww_zzzz |
| audioTrackUID | ATU_vvvvvvvv |

The yyyy part is a four-digit hexadecimal number that represents the **type** of element it is, by using the typeLabel values. Currently there are 5 defined type label values and the possibility to define user custom types:

TABLE 53

**typeDefinitions**

| typeDefinition | typeLabel | Description |
|---|---|---|
| DirectSpeakers | 0001 | For channel-based audio, where each channel feeds a speaker directly |
| Matrix | 0002 | For channel-based audio where channels are matrixed together, such as Mid-Side, Lt/Rt |
| Objects | 0003 | For object-based audio where channels represent audio objects (or parts of objects), so include positional information |
| HOA | 0004 | For scene-based audio where Ambisonics and HOA are used |
| Binaural | 0005 | For binaural audio, where playback is over headphones |
| User Custom | 1yyy to Fyyy | For user custom types. |

The xxxx part is a four-digit hexadecimal number, which identifies the description within a particular type. Values in the range 0001-0FFF are reserved for common definition such as ‘FrontLeft’ or ‘Stereo’. Common definitions are specified in Recommendation ITU-R BS.2094 [8]. Values in the range 1000-FFFF are for custom definitions, which will be particularly used in object-based audio where all the objects will be custom definitions.

The audioChannelFormatID values in the range 0001-0FFF specify the channel with respect to the channel label and channel configuration. The set of defined common definitions for audioChannelFormatIDs for typical speaker positions is found in ITU-R BS.2094 [8]. Some examples of these common definitions are shown in Table 54.

TABLE 54

**Examples of common definition channel labels**

| Sub-element | ID of channel | Name of channel | SpeakerLabel |
|---|---|---|---|
| audioChannelFormatID | AC_00010001 | FrontLeft | M+030 |
| audioChannelFormatID | AC_00010002 | FrontRight | M-030 |
| audioChannelFormatID | AC_00010003 | FrontCentre | M+000 |
| audioChannelFormatID | AC_00010004 | LowFrequencyEffects | LFE |
| audioChannelFormatID | AC_00010005 | SurroundLeft | M+110 |
| audioChannelFormatID | AC_00010006 | SurroundRight | M-110 |

The audioPackFormatID specifies the channel configuration. The set of defined common definitions for audioPackFormatIDs for typical speaker configurations is found in ITU-R BS.2094 [8]. Some examples of the common definitions are shown in Table 55:

TABLE 55

**Examples of common definition for audioPackFormat**

| Sub-element | ID of pack | Name of pack |
|---|---|---|
| audioPackFormatID | AP_00010002 | Stereo_(0+2+0) |
| audioPackFormatID | AP_00010003 | 5.1_(0+5+0) |

In audioBlockFormat the zzzzzzzz part is an 8-digit hexadecimal number that acts as an index/counter for the blocks within the channel. This index should start at 1 for the first block. The yyyyxxxx values should match those of the parent audioChannelFormat ID.

In audioTrackFormat the zz part is a 2-digit hexadecimal number that acts as an index/counter for the tracks within the stream. The yyyyxxxx values should match those of the reference audioStreamFormat ID.

The audioProgramme, audioContent, audioObject and alternativeValueSet do not have a type and so have no yyyy values. As there is initially no intention to have common definitions for these elements the values for wwww will be in the hexadecimal range 1000-FFFF because they will always be custom values. However, keeping the common range of values (0000-0FFF) set aside for now may be useful in future; for example, EBU R 123 configurations may use them.

IDs with a zero value should not be used for any definitions, as they are reserved for elements that should be ignored and are undefined. For example, AT_00000000_00 is for an audioTrackFormat that has no definition and should be ignored. This can be useful for audio files that contain unused tracks (e.g. an 8-track file containing 5-channel audio), so the <chna> chunk can reference AT_00000000_00 in the audioTrackFormat fields for those unused tracks.

Both upper and lower-case hex digits (a-f and A-F) must be supported when reading IDs. Therefore, IDs with the same digits, but with a different case are treated to be identical. For example, AC_0001000a and AC_0001000A are the same ID.

## 7 <chna> Chunk

While the ADM is designed to be a general model, its relationship with the BW64 file specified in Recommendation ITU-R BS.2088 is important to explain. The following describes how a BW64 file does access the ADM metadata via a new RIFF chunk called <*chna*>. An overview of this new chunk is given here.

The ADM is linked to the BW64 file using the audioTrackFormat, audioPackFormat and audioObject (via audioTrackUID) elements. The BW64 file defines a new chunk called <*chna*> (short for ‘channel allocation’), which contains a set of IDs for each track in the file. These IDs either refer to elements, or be referred to from an element.

Each track in the chunk contains the following IDs:

- **audioTrackFormatID** – the ID of the description of a particular audioTrackFormat element. As audioTrackFormat also refers to audioStreamFormat and either audioPackFormat or audioChannelFormat, this ID is enough to describe the format for a particular track. For PCM audio, the audioTrackFormat and the audioStreamFormat may be omitted. Then the same number is used for the ‘yyyyxxxx’ parts of an audioTrackFormat (AT_yyyyxxxx_zz), an audioStreamFormat (AS_yyyyxxxx) and an audioChannelFormat (AC_yyyyxxxx). In this case, when the audioTrackFormat and the audioStreamFormat are omitted, the audioChannelFormatID is referenced by the <chna> chunk in the BW64 file.

- **audioPackFormatID** – the ID of the description of a particular audioPackFormat. As most audioChannelFormats need to be assigned to an audioPackFormat (e.g. ‘FrontLeft’ channel in ‘5.1’ pack), it must be specified in the <*chna*> chunk with this ID.
- **audioTrackUID** – the unique ID that identifies the track. The content descriptor audioObject requires knowledge of which tracks in the file are being described, so contains a list of audioTrackUID references which correspond to audio tracks in the file.

The typeDefinition that audioPackFormatID references does not have to match the typeDefinition that the audioTrackFormatID references for each track. A situation where they may differ is when an encoding matrix definition is being used, where the audioTrackFormatIDs will refer to the ‘DirectSpeakers’ input channels to the matrix, and the audioPackFormatID will refer to ‘Matrix’ type encoding matrix pack.

To enable tracks to contain more than one audioTrackFormatID, in order to allow different formats in the track at different times, the track number can be allocated multiple IDs. An example of such as allocation is below:

TABLE 56

**<chna> chunk example**

| Track No | audioTrackUID | audioTrackFormatID | audioPackFormatID |
|---|---|---|---|
| 1 | 00000001 | 00010001_01 | 00010001 |
| 2 | 00000002 | 00031001_01 | 00031001 |
| 2 | 00000003 | 00031002_01 | 00031002 |

Here, track number two has two audioTrackUIDs as the audioTrackFormats and audioPackFormats assigned to it are used at different times in the file. The times of allocation would have to be found be inspecting the audioObject elements that cover those audioTrackUIDs. An example of this is a programme where tracks 1 and 2 contain the theme tune which lasts for the first minute of the file. These tracks are free after this first minute, so some audio objects from the main body of the programme are stored in them subsequently. As the theme tune and the audio objects have completely different formats and contents they require different audioTrackUIDs.

## 8 Coordinate system

The position elements in audioBlockFormat, for both the ‘DirectSpeakers’ and ‘Objects’ typeDefinitions, allow different axes to be specified in the coordinate attribute. A polar coordinate system, which uses azimuth, elevation and distance is used. The azimuth and elevation angle may also be used for the equation sub-element for scene-based audio (c.f. 5.4.3.4). To ensure consistency when specifying positions each of the polar axes should be based on these guidelines:

– The origin is in the centre, where the sweet-spot would be (although some systems do not have a sweet-spot, so the centre of the space should be assumed).

– Azimuth – angle in the horizontal plane with 0 degrees as straight ahead, and positive angles to the left (or anti-clockwise) when viewed from above.

– Elevation – angle in the vertical plane with 0 degrees horizontally ahead, and positive angles going up.

– Distance – a normalized distance, where 1.0 is assumed to be the default radius of the sphere.

Cartesian coordinates, which is also used for object-based audio, and is supported by using X, Y and Z as the coordinate attributes. It is recommended that normalized values be used here, where the values 1.0 and −1.0 are on the surface of the cube, with the origin being the centre of the cube.

The direction of each axis should be:

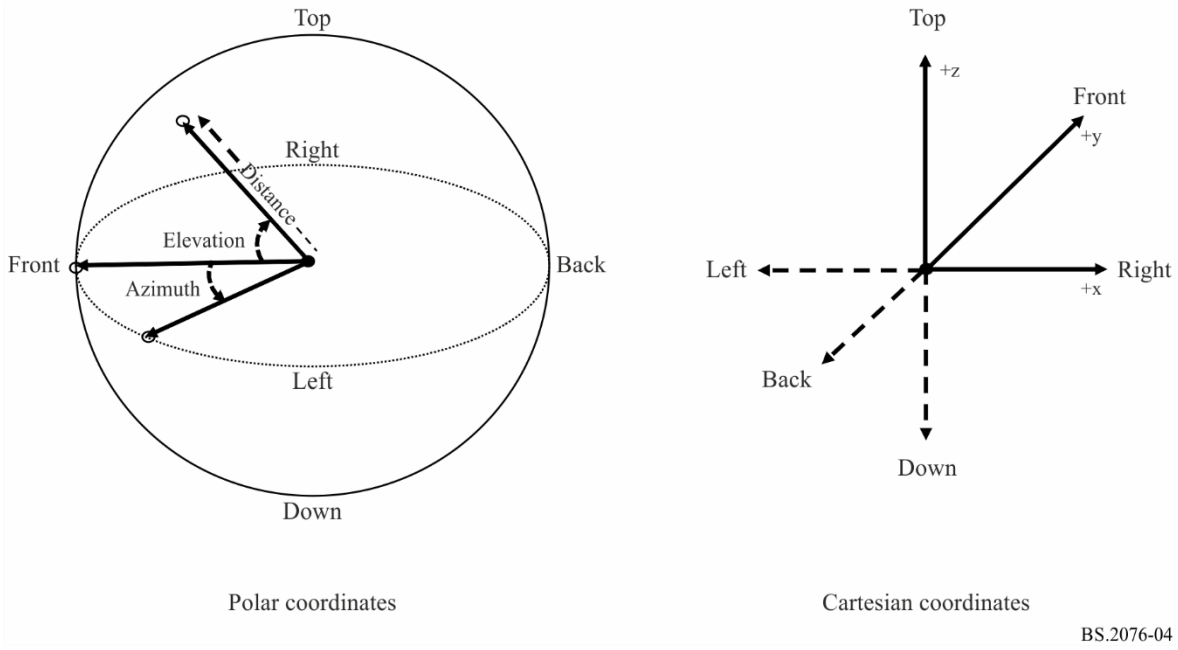
– **X** – left to right, with positive values to the right.

– **Y** – front to back, with positive values to the front.

– **Z** – top to bottom, with positive values to the top.

FIGURE 4

**Coordinate systems used for Objects**




If normalized distances are used in the coordinate system they can be scaled to an absolute distance by multiplying by the absoluteDistance parameter in the audioPackFormat.

For scene-based audio, the coordinate system is also Cartesian based, but the axes are different. The reason for the different axes for scene-based audio is a legacy of the development of Ambisonics, which has always used these axes. In this case the direction of each axis is:

– **X** – front to back, with positive values to the front.

– **Y** – left to right, with positive values to the left.

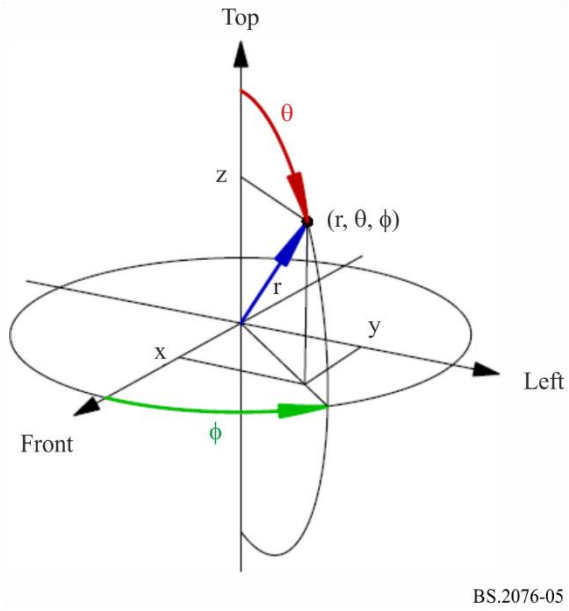
– **Z** – top to bottom, with positive values to the top.

To avoid confusion with the other Cartesian system, it is recommended the axes be labelled ‘X_HOA’, ‘Y_HOA’ & ‘Z_HOA’. However, the HOA component definitions are unlikely to include coordinate information and so this information is primarily to ensure the rendering is correctly done.

The spherical coordinate system for scene-based audio is used according to Fig. 5.

FIGURE 5

**Spherical and Cartesian coordinate system as used for HOA**




## 9 Common parameter descriptions for all typeDefinitions

There are four parameters that are common to all the typeDefinitions:

– **importance** defined in audioBlockFormat, audioPackFormat and audioObject;

– **gain** occurs in both audioBlockFormat and audioObject;

– **headLocked** occurs in both audioBlockFormat and audioObject;

– **headphoneVirtualise** occurs in audioBlockFormat.

### 9.1 gain

The **gain** parameter is a linear or logarithmic gain and controls the level of the referenced audio signal. At rendering/playback the level of signal will be multiplied by the gain value. If the gain parameter is not set, a value of 1.0 is assumed, so the audio signal's level is not adjusted.

Ideally, the waveform that is being described should be at the desired level, so the gain parameter is not required (or set to 1.0), rather than relying on the gain parameter to adjust levels.

For a detailed description of the relationship and application of gain parameters in the ADM, see § 12.

### 9.2 importance

The **importance** parameter allows a processor to compromise audio tracks below a certain level of importance, with 10 being the most important, and 0 the least. This parameter for example may be useful when the size of the ADM metadata needs to be reduced and allow prioritisation to be made on what compromises can be made.

When the importance parameter is used in audioObject it can be used to remove less important sounds when the number of objects or tracks needs to be reduced. For example, some background sound effects can be discarded to ensure main dialogue objects are retained.

When the importance parameter is used in audioPackFormat it can be used to compromise on spatial audio quality. Nested audioPackFormats can be used to exploit this feature. For example, an audio object with a main direct sound (in a parent audioPackFormat with high importance) and additional

reverb sounds (in a child audioPackFormat with low importance), could have the reverb sound discarded which retains the main sound, but compromises quality.

The importance parameter in audioBlockFormat can be used in a similar way to audioPackFormat to allow spatial quality to be compromised, but care must be taken that the sound is not adversely repositioned as a result of discarding channels.

### 9.3 headLocked

The **headLocked** flag indicates that an audio object should lock to the listener's head when the head is moved (yaw/pitch/roll). Therefore, a headphone renderer which uses head tracking should not track the object if headLocked is set to "1". Figure 6 depicts the concept of enabled and disabled head-locked audio elements.

The default state (when headLocked is not present) is for head-locking to be off, so the scene of objects remains fixed relative to the moving head (the middle diagram in Fig. 6).

If **headLocked** is present in the audioOject and the audioBlockFormat, the value defined in the **audioBlockFormat** shall take precedence over the audioObject value.

### 9.4 headphoneVirtualise

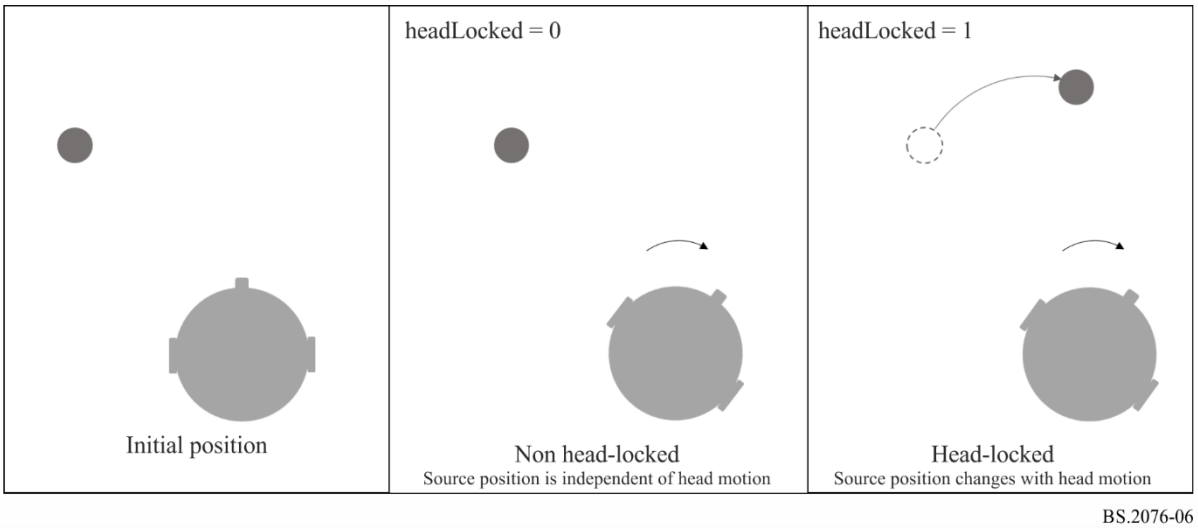
The **headphoneVirtualise** element specifies whether the content of the audioChannelFormat should be rendered with headphone virtualisation. The element consists of two attributes: "**bypass**" and "**DRR**" (Direct-to-Reverberant ratio).

The **bypass** attribute is a 1/0 flag that signals whether the content should be rendered using a headphone virtualiser (value of 0) or renderer to stereo (value of 1).

The **DRR** attribute defines the Direct-to-Reverberant-Ratio (DRR) in dB. This can be given in the range of −130 dB to 130 dB with 130 dB meaning anechoic (all direct sound).

FIGURE 6

**Intended behaviour for head-locked audio elements**




## 10 Parameter descriptions for typeDefinition of 'Objects'

These parameters are found in the audioBlockFormat when the typeDefinition is 'Objects'.

### 10.1 diffuse

The **diffuse** value between 0.0 and 1.0 describes the diffuseness of a sound, where 0.0 (the default) is a direct non-diffuse sound, and 1.0 a completely diffuse sound.

### 10.2 channelLock

If the **channelLock** flag is set to 1 then the renderer will send the audio signal to the nearest (in terms of 3D position) channel or speaker position. A typical application for this is where the exact location of the object is not critical, but the need for un-processed reproduction of that signal takes priority.

The optional maxDistance attribute defines the radius $r$, $0 \leq r \leq 2$, of a sphere around the object's position. If one or more speakers exist in the defined sphere or on its surface, the object snaps to the nearest speaker. If maxDistance is undefined, a default value of infinity is assumed, meaning that the object should snap to the nearest of all speakers (unconditioned channelLock).

### 10.3 jumpPosition and interpolationLength

If the **jumpPosition** flag is set to 0 then the renderer will interpolate a moving object between positions over the full duration of the block. If it is set to 1 it will jump to the new position instantly. If the **interpolationLength** attribute is used when **jumpPosition** is 1, then the interpolation period is set to the **interpolationLength** value. The **interpolationLength** should be no longer than the block's duration.

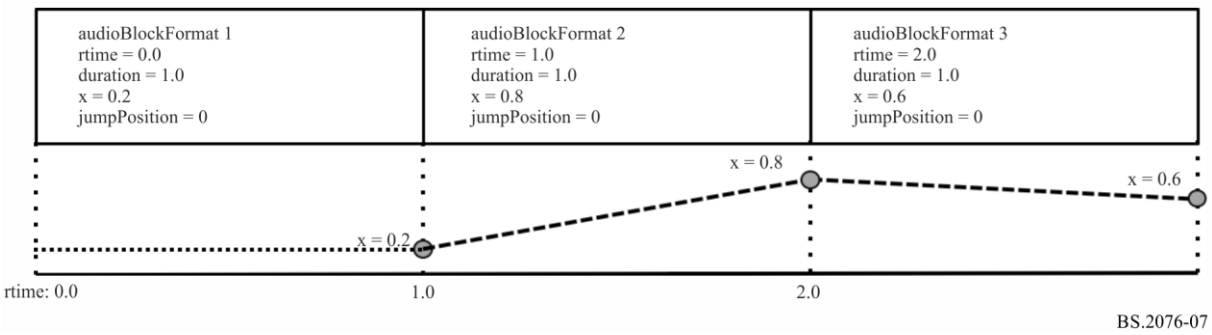
The **interpolationLength** parameter allows the interpolation of a moving object to be done over a shorter time period than the next update time. This allows the control of the crossfading of objects that may be desirable due to processing done to objects. If the value is set to zero then the object will jump position without interpolation. If this attribute is not included when jumpPosition is set to 1, then the interpolation length will be set to 0.

It is recommended that audioBlockFormat sizes are chosen to be small enough to avoid the use of the **interpolationLength** parameter for smoothly moving objects.

To help illustrate how jumpPosition and interpolationLength are interpreted, the following diagrams show a sequence of audioBlockFormats and how a dynamic parameter's value varies over time. The first example, in Fig. 7, shows when jumpPosition is set to zero (or not used), so the parameter (arbitrary parameter 'x' in this case) is interpolated over the duration of the entire audioBlockFormats. As the first block has a jumpPosition of zero and is not proceeded by another block the x value is only known at the end of the block, therefore the position at the start of the first block is effectively undefined. If this situation occurs, then the position at the start of the first block is made the same as the end of the block.

FIGURE 7

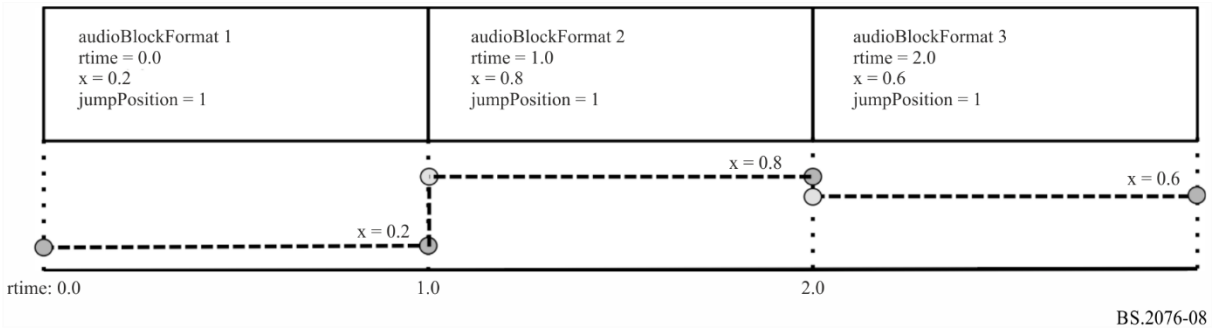
**Interpolation with no jumpPosition**



The second example, in Fig. 8, shows how the value of x varies when jumpPosition is set to 1 and no interpolationLength is set. The value of x is set at the beginning of the block and maintains that value throughout its duration. This also shows that the first block has a defined position from the beginning, and thus illustrates that it is recommended to set jumpPosition to 1 for the first block in a sequence.

FIGURE 8

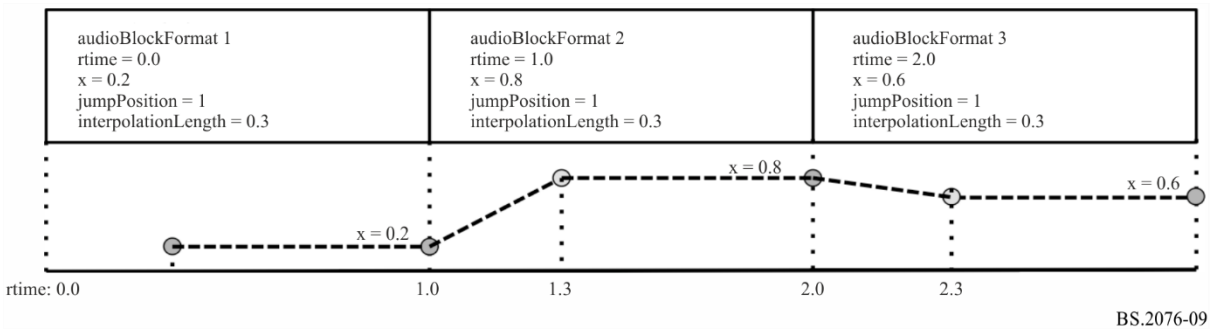
**Interpolation with jumpPosition set**




The third example, in Fig. 9, shows how the use of the interpolationLength attribute varies the value of x over the sequence of blocks. In this example, each interpolationLength is set to 0.3, so the value of x is interpolated over the first 0.3 seconds of the block, and then is locked to the defined value for the remainder of the block. The first block has an undefined value of x for the first 0.3 seconds.

FIGURE 9

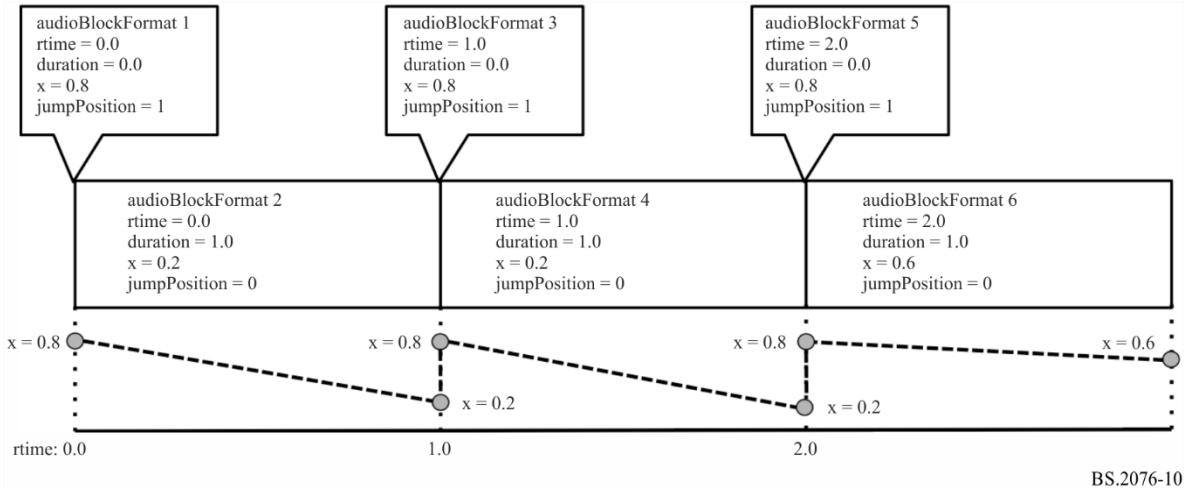
**Interpolation with interpolationLength used with jumpPosition**




The fourth example, in Fig. 10, shows how zero length blocks can be used to make a position jump, but also allow for interpolation to follow immediately. By having a first block of zero length it can ensure an initial position is always present.

FIGURE 10

**Interpolation using zero length blocks**




To ensure undefined behaviour of the first block is avoided, then the position specified in the first block covers the entire length of the block (regardless of the jumpPosition and interpolationLength properties).

The following parameters can be interpolated: position, width, height, depth, diffuse, gain, and objectDivergence.

The other parameters in audioBlockFormat should not be interpolated and should remain constant for the duration of the block.

### 10.4 zoneExclusion

The **zoneExclusion** parameter is used to dynamically reconfigure the object renderer to “mask out” certain speaker zones during playback. This guarantees that no loudspeaker belonging to the masked zones will be used for rendering the applicable object. Typical zone masks used in production today include sides and rear. Multiple **zone** sub-elements within **zoneExclusion** can be set simultaneously to mask out more than one zone. The default is that all zones are enabled and when zoneExclusion is set to one or more of the indicated zones, those are “masked out” during playback. The sub-element **zone** is used to define the coordinates of the zone in the unit-cuboid.

Zones are defined in the Cartesian coordinate system using the sub-element **zone** by specifying the corner points of a unit-cuboid in 3D space by: minX, maxX, minY, maxY, minZ, maxZ. In the spherical coordinate system the zone is defined by: minAzimuth, maxAzimuth, minElevation, maxElevation.

For example: minX= −1.0, maxX=1.0, minY= −1.0, maxY= –1.0, minZ= −1.0, maxZ=1.0 specifies the *rear* wall.

### 10.5 objectDivergence

The **objectDivergence** parameter (0.0 to 1.0) indicates the amount an object is split symmetrically into a pair of virtual objects, so that a phantom object is created in the position of the original object. The spread of the signal between the virtual objects should not create an image shift from the original object position and should be power preserving across virtual objects and the original. The **azimuthRange** and **positionRange** attributes allow the relative positions of virtual objects to be specified. This can either be an angle where spherical coordinates are being used, or a distance value where Cartesian coordinates are being used. When spherical coordinates are used, a value of 45 degrees would place virtual objects 45 degrees to the left and right of the specified object. The

default angle is 0 degrees if this attribute is not used. When Cartesian coordinates are used, a value of 0.5 would place the virtual objects at x-0.5,y,z and x+0.5,y,z if x,y,z is the location of the specified object. The default distance is 0.0.

The values of **objectDivergence** should be interpreted as:

TABLE 57

**objectDivergence values**

| Value | Description |
|---|---|
| 0 | No divergence with only the original object being present. |
| 1 | Maximum divergence where this would represent virtual objects being created azimuthRange degrees on either side of the original position. |

Example: With an LCR loudspeaker layout and the object positioned directly at the C position, and the LR virtual objects specified by using an **azimuthRange** of 30 degrees. An **objectDivergence** value of 0 indicating no divergence, only the centre speaker would be firing. A value of 0.5 would have all three (LCR) loudspeakers firing equally, and a value of 1 would have the L and R loudspeakers firing equally.

### 10.6 screenRef and audioProgrammeReferenceScreen

The **screenRef** flag is used to indicate whether the corresponding audio signal (e.g. object or HOA signal) is screen-related or not. The screenRef flag can be used by a renderer for special processing of all screen-related objects taking into account the size of a local reproduction screen compared to the production screen-size.

If a renderer uses the screenRef flag to enable a special processing, it should use the reference/monitoring/production screen-size of the currently rendered audioProgramme as the reference screen.

If the flag is set and no audioProgrammeReferenceScreen element is included in the corresponding currently rendered audioProgramme, the reference production/monitoring screen is implicitly defined on the basis of Recommendation ITU-R BT.1845 – Guidelines on metrics to be used when tailoring television programmes to broadcasting applications at various image quality levels, display sizes and aspect ratios [6].

TABLE 58

**Default screen size**

| | |
|---|---|
| Azimuth of left bottom corner of screen | 29.0° |
| Elevation of the left bottom corner of screen | −17.3° |
| Aspect ratio | 1.78 (16:9) |
| Polar angular width of the screen | 58°<br>(as defined by image system<br>3840 × 2160) |

These spherical values can be transferred to Cartesian coordinates assuming a reference distance of 1.0 by first transferring the values above to the “standard” azimuth/elevation convention (0° azimuth is in front of the right ear, positive values are counted counter-clockwise; 0° elevation is directly above the head, positive values are counted downwards to the front) and then using the trigonometric

functions to gain the Cartesian coordinates. The screen is assumed to have its centre touching the unit sphere. This results in the following values (orientation of the Cartesian coordinate axes as in § 8):

TABLE 59

**Default screen size in Cartesian coordinates**

| | |
|---|---|
| X-coordinate of the centre of the screen | 0.0 |
| Y-coordinate of the centre of the screen | 1.0 |
| Z-coordinate of the centre of the screen | 0.0 |
| Aspect ratio | 1.78 |
| Width of the screen | 1.1086 |

NOTE – The maths to convert from the polar coordinate screen to Cartesian coordinates is:

– $d = \frac{1}{\sqrt{\left(\frac{1}{a^2}\right)\tan^2\left(\frac{w}{2}\right)+1}}$

where *d* is the Y-coordinate of the centre of the screen, *a* is the aspect ratio, and *w* is the polar angle of the screen width.

– $x = 2d\tan(\frac{w}{2})$

where *x* is the Cartesian screen width, and *w* is the polar angle of the screen width.

## 11 Parameter descriptions for typeDefinition of 'HOA'

These parameters are found in the audioBlockFormat when the typeDefinition is 'HOA'.

### 11.1 order and degree

The meaning of **order** and **degree** values is based on the following definition of real-valued spherical harmonics:

$$Y_n^m(\theta,\phi) = N_n^{|m|}P_n^{|m|}(\cos(\theta))\begin{cases}\sqrt{2}\cos(m\phi), & \text{for } m > 0\\ 1, & \text{for } m = 0\\ -\sqrt{2}\sin(m\phi), & \text{for } m < 0\end{cases}$$

where:

$n$ : order value

$m$ : degree value

$\phi$ : azimuth

$\theta$ : elevation

$N_n^{|m|}$ : normalization parameter for the given order and degree

$P_n^{|m|}$ : associated Legendre function for the given order and degree.

The associated Legendre functions $P_n^m(x)$ are defined as:

$$P_n^m(x) = (1-x^2)^{\frac{m}{2}}\frac{\mathrm{d}^m}{\mathrm{d}x^m}P_n(x),\ m \geq 0$$

with the Legendre polynomial $P_n(x)$ and without the Condon-Shortley phase term $(-1)^m$.

### 11.2 normalization

When the **normalization** is specified as N3D, the following equation is given:

$$N_{\mathrm{N3D}_n^{|m|}} = \sqrt{(2n+1)\frac{(n-|m|)!}{(n+|m|)!}}.$$

N3D normalization yields a set of orthonormal basis functions. With N3D normalization the higher-order components ($n \geq 0$) can have an energy greater than that of the $n = 0$ component, which risks causing clipping distortions when audio data is stored in integer sample formats.

When the **normalization** is specified as SN3D, the following equation is given:

$$N_{\mathrm{SN3D}_n^{|m|}} = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}}.$$

SN3D normalization applies a weighting to the HOA components according to the order such that the energy does not exceed that of the $n = 0$ component.

When the **normalization** is specified as FuMa, the signal was stored with the Furse-Malham (FuMa) weighting. This system of weighting is designed for coefficients not to exceed an absolute value of 1 in panning. It also has a −3 dB weighting of the *n* = 0 component. It is only defined up to order 3.

TABLE 60

**HOA FuMa normalization**

| **Order ($n$)** | **Degree ($\|m\|$)** | $N_{\mathrm{FuMa}_n^{\|m\|}}$ ***Normalization (relative to*** $N_{\mathrm{SN3D}_n^{\|m\|}}$***)*** |
|---|---|---|
| 0 | 0 | $\frac{1}{\sqrt{2}} N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 1 | 0 | $N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 1 | 1 | $N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 2 | 0 | $N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 2 | 1 | $\frac{2}{\sqrt{3}} N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 2 | 2 | $\frac{2}{\sqrt{3}} N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 3 | 0 | $N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 3 | 1 | $\sqrt{\frac{45}{32}} N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 3 | 2 | $\frac{3}{\sqrt{5}} N_{\mathrm{SN3D}_n^{\|m\|}}$ |
| 3 | 3 | $\sqrt{\frac{8}{5}} N_{\mathrm{SN3D}_n^{\|m\|}}$ |

To reduce the risk of clipping with integer sample formats the SN3D normalization is the default option. Due to its greater dynamic range, N3D normalization is recommended for floating-point sample formats where there is practically no risk of clipping.

### 11.3 nfcRefDist

The **nfcRefDist** indicates the reference distance (in metre) that has been used during the scene-based audio production. This reference distance may be used for the audio rendering for nearfield compensation (NFC) [9].

If the **nfcRefDist** is not defined or set to zero, nearfield-compensated rendering is not intended.

### 11.4 screenRef

The **screenRef** flag is used to indicate whether the scene-based programme is screen-related or not.

The screenRef flag can be used by a renderer for special adaptation of the scene-based content taking into account the size of a local reproduction screen in relation to the production screen-size.

See § 10.6 for additional information regarding the production screen-size parameter.

### 11.5 Ambisonics Channel Numbering

An often-used convention for channel ordering based on order and degree components is the so-called Ambisonics Channel Number (ACN):

$$\mathrm{ACN} = n^2 + n + m.$$

The order and degree components can be easily retrieved from the ACN number:

$$n = \left\lfloor \sqrt{\mathrm{ACN}} \right\rfloor,$$

$$m = \mathrm{ACN} - n^2 - n.$$

## 12 Relationship and application of gain parameters in the ADM

The following elements of the ADM are relevant to calculate the final gain of a specific audio sample:

– **Gain sub-element of audioBlockFormat**: Defines a gain value (either linear or logarithmic) that should be applied to all audio samples corresponding to the parent audioBlockFormat. If the gain parameter is not set, a linear value of 1.0 is assumed. Ideally the waveform (represented e.g. by PCM samples) should be at the desired level, so the gain parameter is not required (or set to 1.0). The gain parameter in audioBlockFormat is useful when a single audio track is being used by multiple audioChannelFormat definitions, each requiring different levels.

– **Gain sub-element of audioObject**: Defines a gain value (either linear or logarithmic) that should be applied to all audio samples corresponding to the parent audioObject. The gain parameter in audioObject can for instance be used for user interactivity. Then it describes the initial playback gain of the audioObject during rendering. For example, it might be required that a particular audioObject is usually muted, so it will be given a gain of zero (-inf dB). It can also be used to ensure that different audioProgrammes using a different combination of audioObjects maintain a desired loudness level. If the gain parameter is not set, a linear value of 1.0 (0 dB) is assumed.

– **gainInteractionRange sub-element of audioObjectInteraction**: The audioObjectInteraction sub-element of audioObject can be used to define in which boundaries a user can interactively influence the audioObject. With relation to gain, it is possible to allow or forbid any gain interaction at all. If gain interaction is allowed, the gainInteractionRange sub-element of audioObjectInteraction defines minimum and maximum boundaries for the gain interaction (either as linear or logarithmic values). Any change to an attribute that can be set by the user should be within the limits of the interaction range.

During rendering/playback, all the different gain parameters and related ADM metadata have to be combined in a specific way to ensure the correct playback level is chosen for specific set of audio samples or an audio source. The combination of the different gain parameters is defined in Figs 11 and 12.

FIGURE 11

**Application of audioObject gain and audioBlockFormat gain (linear values)**

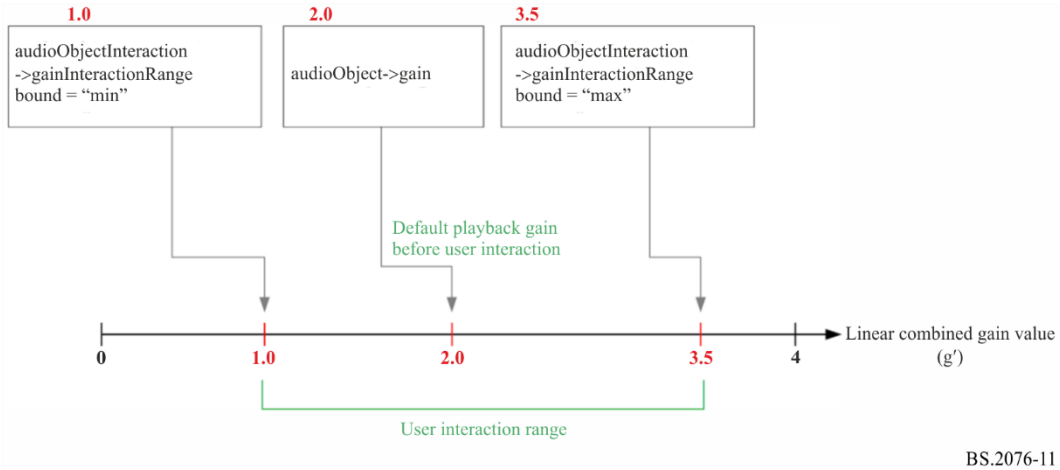
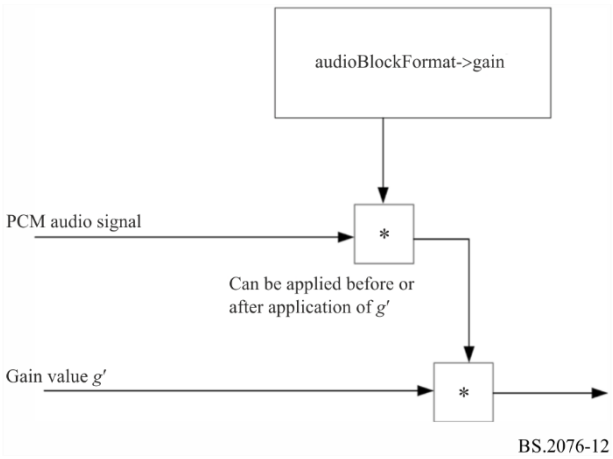



FIGURE 12

**Application of the combined gain (linear values)**




The combined linear gain value g' is calculated as follows:

$$g' = \min\left(g_{max}, \max(g_{min}, g_{User})\right)$$

with

$g_{min}$ …audioObjectInteraction->gainInteractionRange bound="min"

$g_{max}$ …audioObjectInteraction->gainInteractionRange bound="max"

$g_{User}$ …gain value imposed by user interaction

If the user does not change the gain, but keeps the original default playback gain, $g_{User}$ is equal to $g_{AO}$, with $g_{AO}$ being the audioObject->gain value.

The resulting overall playback gain value then is

$$g_{total} = g_{block} \cdot g'$$

with $g_{block}$ being the audioBlockFormat->gain value.

In case the gain parameters are defined by logarithmic values in dB, the multiplication blocks in above block diagram and the multiplications in above formulas have to be replaced by additions to calculate the final overall gain value.

Linear and logarithmic gain values can be translated as follows:

$$g_{log}[dB] = 20 \cdot \log_{10}(g_{lin})$$

$$g_{lin} = 10^{(\frac{g_{log}[dB]}{20})}$$

A linear value of 0 is equivalent to a logarithmic value of negative infinity ("-INF").

## 13 Application of position-related parameters in the ADM

The following elements of the ADM are relevant to calculate the final position to which a specific audio sample should be rendered:

– **position sub-element of audioBlockFormat** (for typeDefinition="DirectSpeakers" and typeDefinition="Objects"): This element defines the position of either a loudspeaker (typeDefinition="DirectSpeakers") or a single sequence of audioChannelFormat samples representing an object. The position can be given either by azimuth, elevation and normalised distance (polar/spherical coordinates) or normalised x, y, z values (Cartesian coordinates).

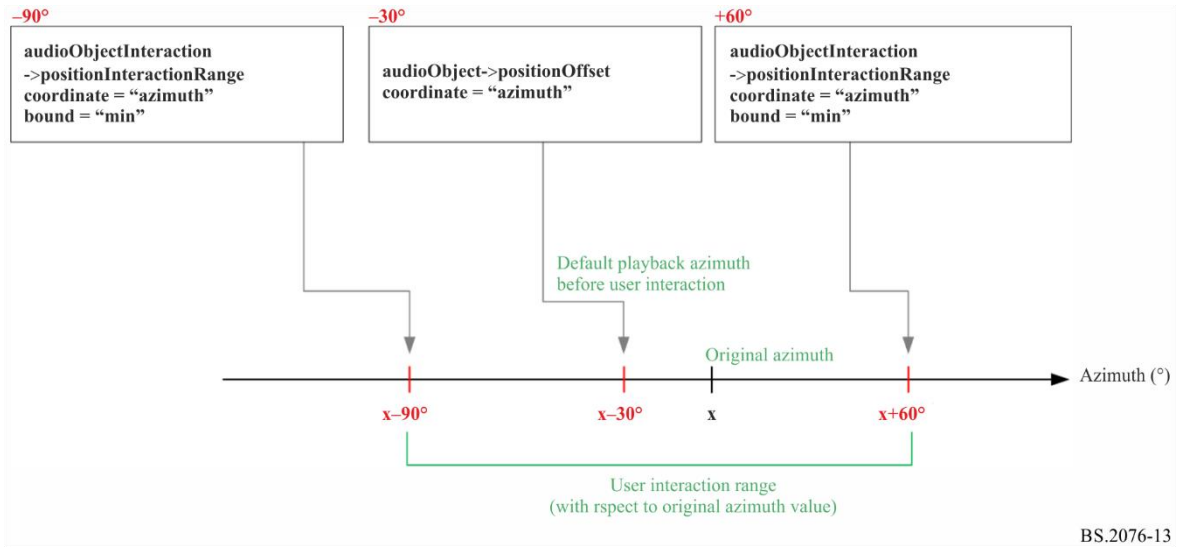
– **positionOffset sub-element of audioObject**: This defines position offset values that should be applied to the position metadata of all audio corresponding to the parent audioObject. It describes the initial playback position offset of the audioObject during rendering.

– **positionInteractionRange sub-element of audioObjectInteraction**: This element defines the bounds in which a user-side position interaction is possible. It gives minimum and maximum values for a possible user interaction with respect to azimuth, elevation and distance (spherical coordinates) or X, Y, Z (Cartesian coordinates).

During rendering/playback, all the different position-related parameters and related ADM metadata have to be combined in a specific way to ensure the correct rendering position is chosen for specific set of audio samples or an audio source. The combination of the different gain parameters is depicted exemplary for the azimuth value of an object in Fig. 13.

FIGURE 13

**Application of audioObject position offset values (polar coordinates)**



## 14 References

[1] Report ITU-R BS.2266 – Framework of future audio broadcasting systems

[2] Recommendation ITU-R BS.1909 – Performance requirements for an advanced multichannel stereophonic sound system for use with or without accompanying picture

[3] Recommendation ITU-R BS.2051 – Advanced sound system for programme production

[4] Recommendation ITU-R BS.1352 – File format for the exchange of audio programme materials with metadata on information technology media

[5] Recommendation ITU-R BS.1770 – Algorithms to measure audio programme loudness and true-peak audio level

[6] Recommendation ITU-R BT.1845 – Guidelines on metrics to be used when tailoring television programmes to broadcasting applications at various image quality levels, display sizes and aspect ratios

[7] Recommendation ITU-R BS.2088 – Long-form file format for the international exchange of audio programme materials with metadata

[8] Recommendation ITU-R BS.2094 – Common definitions for the Audio Definition Model

[9] Daniel J. Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new ambisonic format. In 23rd International AES Conference: Signal Processing in Audio Recording and Reproduction 2003

# Annex 2
# (informative)

# Examples of ADM usage

This Annex 2 contains a selection of examples of metadata that uses the ADM. These are to help illustrate how the ADM is used, but should not be considered as references for audio definitions.

## 1 Channel-based example

The most common use of audio is still channel-based, where tracks within a file each represent a static audio channel. This example demonstrates how to define two tracks, streams and channels; and a pack for stereo. The track and stream definitions are for PCM audio. Two objects are defined, both stereo, but containing different content so there are 4 tracks used. This example uses a programme called ‘Documentary’ containing ‘Music’ and ‘Speech’ each defined as separate stereo objects.

The format-related elements in this example represent a tiny subset of the common reference set of definitions. In practice, this XML code would be part of the common reference file and would not have to be included in the BWF file. All that would be required is a <*chna*> chunk with the references to the audioTrackFormats and audioPackFormats and any extra XML required for audioObject, audioContent and audioProgramme.

### 1.1 Summary of elements

These are the elements in the format part of the description:

TABLE 61

**Channel-based example format elements**

| Element | ID | Name | Description |
|---|---|---|---|
| audioTrackFormat | AT_00010001_01 | PCM_FrontLeft | Defines track as PCM |
| audioTrackFormat | AT_00010002_01 | PCM_FrontRight | Defines track as PCM |
| audioStreamFormat | AS_00010001 | PCM_FrontLeft | Defines stream as PCM |
| audioStreamFormat | AS_00010002 | PCM_FrontRight | Defines stream as PCM |
| audioChannelFormat & audioBlockFormat | AC_00010001 AB_00010001_00000001 | FrontLeft | Describes channel as front left with a position and speaker reference |
| audioChannelFormat & audioBlockFormat | AC_00010002 AB_00010002_00000001 | FrontRight | Describes channel as front right with a position and speaker reference |
| audioPackFormat | AP_00010002 | Stereo | Defines a stereo pack referring to two channels. |

These are the elements in the content part of the description:

TABLE 62

**Channel-based example content elements**

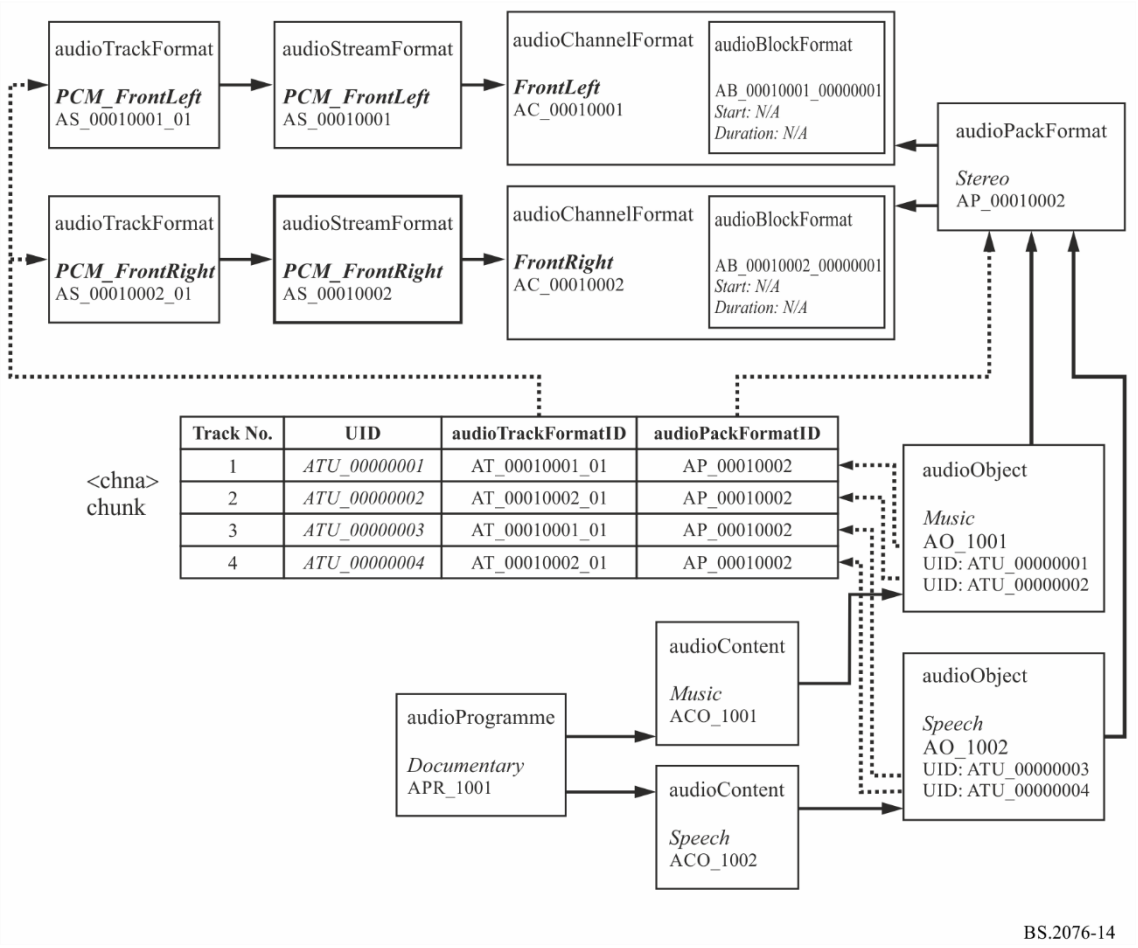
| Element | ID | Name | Description |
|---|---|---|---|
| audioObject | AO_1001 | Music | Object for ‘Music’, stereo format |
| audioObject | AO_1002 | Speech | Object for ‘Speech’, stereo format |
| audioContent | ACO_1001 | Music | Music content |
| audioContent | ACO_1002 | Speech | Speech content |
| audioProgramme | APR_1001 | Documentary | Programme ‘Documentary’ containing ‘Music’ and ‘Speech’ content |

## 1.2 Element Relationships

The diagram in Fig. 14 shows how the defined elements relate to each other. The top half of the diagram covers the elements that describe the two-channel stereo format. The <*chna*> chunk in the middle shows how the four tracks are connected to the format definitions. The content definition elements are at the bottom of the diagram, with the audioObject elements containing the track UID references to the UIDs in the <*chna*> chunk.

FIGURE 14

**Channel-based example diagram**




## 1.3 Sample code

This XML sample code does not include the audioFormatExtended parent element and the XML header for clarity.

The first excerpt of code covers the format elements, which could be contained within the common definitions reference file:

```
<!-- ############ -->
<!-- PACKS -->
<!-- ############ -->

<audioPackFormat audioPackFormatID="AP_00010002" audioPackFormatName="Stereo"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
</audioPackFormat>

<!-- ############ -->
<!-- CHANNELS -->
<!-- ############ -->

<audioChannelFormat audioChannelFormatID="AC_00010001"
audioChannelFormatName="FrontLeft" typeLabel="0001" typeDefinition="DirectSpeakers">
```

```
  <audioBlockFormat audioBlockFormatID="AB_00010001_00000001">
    <speakerLabel>M+030</speakerLabel>
    <position coordinate="azimuth">30.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010002"
audioChannelFormatName="FrontRight" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010002_00000001">
    <speakerLabel>M-030</speakerLabel>
    <position coordinate="azimuth">-30.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<!-- ############ -->
<!-- STREAMS -->
<!-- ############ -->

<audioStreamFormat audioStreamFormatID="AS_00010001"
audioStreamFormatName="PCM_FrontLeft" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010002"
audioStreamFormatName="PCM_FrontRight" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
</audioStreamFormat>

<!-- ############ -->
<!-- AUDIO TRACKS -->
<!-- ############ -->

<audioTrackFormat audioTrackFormatID="AT_00010001_01"
audioTrackFormatName="PCM_FrontLeft" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010001</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010002_01"
audioTrackFormatName="PCM_FrontRight" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010002</audioStreamFormatIDRef>
</audioTrackFormat>
```

The second excerpt covers the content part, which would have to be included in the <*axml*> chunk of the BWF file:

```
<!-- ############ -->
<!-- PROGRAMMES -->
<!-- ############ -->

<audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Documentary">
  <audioContentIDRef>ACO_1001</audioContentIDRef>
  <audioContentIDRef>ACO_1002</audioContentIDRef>
```

```
</audioProgramme>

<!-- ############ -->
<!-- CONTENTS -->
<!-- ############ -->

<audioContent audioContentID="ACO_1001" audioContentName="Music">
  <audioObjectIDRef>AO_1001</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-28.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<audioContent audioContentID="ACO_1002" audioContentName="Speech">
  <audioObjectIDRef>AO_1002</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-23.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<!-- ############ -->
<!-- OBJECTS -->
<!-- ############ -->

<audioObject audioObjectID="AO_1001" audioObjectName="Music" start="00:00:00.00000">
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
</audioObject>

<audioObject audioObjectID="AO_1002" audioObjectName="Speech" start="00:00:00.00000">
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000003</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000004</audioTrackUIDRef>
</audioObject>

<!-- ############ -->
<!-- AUDIO TRACK UIDs -->
<!-- ############ -->

<audioTrackUID UID="ATU_00000001">
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000002">
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000003">
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000004">
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
</audioTrackUID>
```

## 2 Object-based example

To demonstrate how the ADM can be used in object-based audio here is a simple example using a single object. This example uses multiple audioBlockFormats within an audioChannelFormat to describe the dynamic properties of an object called ‘Car’. The audioBlockFormats uses the start and duration attributes to frame the time dependent metadata, thus allowing the object’s position to move in space.

### 2.1 Summary of elements

These are the elements in the format part of the description:

TABLE 63

**Object-based example format elements**

| Element | ID | Name | Description |
|---|---|---|---|
| audioTrackFormat | AT_00031001_01 | PCM_Car1 | Defines track as PCM |
| audioStreamFormat | AS_00031001 | PCM_Car1 | Defines stream as PCM |
| audioChannelFormat & audioBlockFormat | AC_00031001<br>AB_00031001_00000001<br>AB_00031001_00000002<br>AB_00031001_00000003 | Car1 | Describes channel as an object type containing three blocks with different positional metadata in each. |
| audioPackFormat | AP_00031001 | Car | Defines a pack referring to one channel. |

These are the elements in the content part of the description:

TABLE 64

**Object-based exmaple content elements**

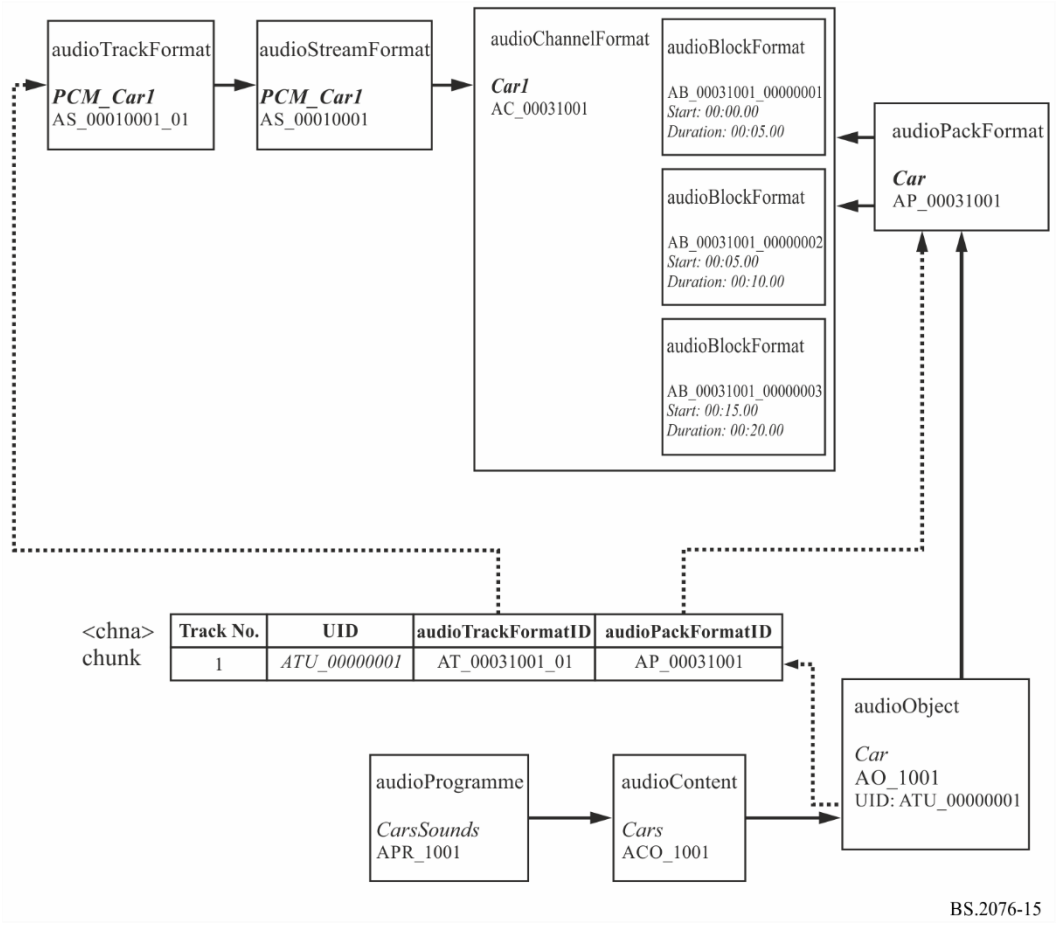
| Element | ID | Name | Description |
|---|---|---|---|
| audioObject | AO_1001 | Car | Object for ‘Car, stereo format |
| audioContent | ACO_1001 | Cars | ‘Cars’ content |
| audioProgramme | APR_1001 | CarsSounds | Programme ‘CarsSounds’ containing ‘Cars’ content |

### 2.2 Element Relationships

The diagram in Fig. 15 shows how the defined elements relate to each other. The top half of the diagram covers the elements that describe the single channel object containing three blocks. The <*chna*> chunk in the middle shows how the single track is connected to the format definitions. The content definition elements are at the bottom of the diagram, with the audioObject element containing the track UID references to the UID in the <*chna*> chunk.

FIGURE 15

**Object-based example diagram**




## 2.3 Sample code

This XML sample code does not include the audioFormatExtended parent element and the XML header for clarity. The excerpt of code covers both the format and content elements:

```
<!-- ############ -->
<!-- PROGRAMMES -->
<!-- ############ -->

<audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="CarsSounds">
  <audioContentIDRef>ACO_1001</audioContentIDRef>
</audioProgramme>

<!-- ############ -->
<!-- CONTENTS -->
<!-- ############ -->

<audioContent audioContentID="ACO_1001" audioContentName="Cars">
  <audioObjectIDRef>AO_1001</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-23.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<!-- ############ -->
```

```
<!-- OBJECTS -->
<!-- ############ -->

<audioObject audioObjectID="AO_1001" audioObjectName="Car" start="00:00:00.00000">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
</audioObject>

<!-- ############ -->
<!-- PACKS -->
<!-- ############ -->

<audioPackFormat audioPackFormatID="AP_00031001" audioPackFormatName="Car"
typeLabel="0003" typeDefinition="Objects">
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
</audioPackFormat>

<!-- ############ -->
<!-- CHANNELS -->
<!-- ############ -->

<audioChannelFormat audioChannelFormatID="AC_00031001" audioChannelFormatName="Car1"
typeLabel="0003" typeDefinition="Objects">
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
duration="00:00:05.00000">
    <position coordinate="azimuth">-22.5</position>
    <position coordinate="elevation">5.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:05.00000"
duration="00:00:10.00000">
    <position coordinate="azimuth">-24.5</position>
    <position coordinate="elevation">6.0</position>
    <position coordinate="distance">0.9</position>
  </audioBlockFormat>
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000003" rtime="00:00:15.00000"
duration="00:00:20.00000">
    <position coordinate="azimuth">-26.5</position>
    <position coordinate="elevation">7.0</position>
    <position coordinate="distance">0.8</position>
  </audioBlockFormat>
</audioChannelFormat>


<!-- ############ -->
<!-- STREAMS -->
<!-- ############ -->

<audioStreamFormat audioStreamFormatID="AS_00031001" audioStreamFormatName="PCM_Car1"
formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
</audioStreamFormat>

<!-- ############ -->
<!-- AUDIO TRACKS -->
<!-- ############ -->
```

```
<audioTrackFormat audioTrackFormatID="AT_00031001_01" audioTrackFormatName="PCM_Car1"
formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00031001</audioStreamFormatIDRef>
</audioTrackFormat>
```

## 3 Scene-based example

The other main type of audio is scene-based where the audio channels are representing Ambisonics/HOA components. Their use is very similar to that of the channel-based approach with the main difference being the parameters used within audioBlockFormat. This example shows a simple 1st order Ambisonics (using the N3D normalization) configuration using four channels mapped onto four tracks. Like the channel-based approach, the format elements would be defined in a common reference file so in practice would not need to be included in the BWF file itself.

### 3.1 Summary of elements

These are the elements in the format part of the description:

TABLE 65

**Scene-based example format elements**

| Element | ID | Name | Description |
|---|---|---|---|
| audioTrackFormat | AT_00040101_01 | PCM_N3D_ACN_0 | Defines track as PCM |
| audioTrackFormat | AT_00040102_01 | PCM_N3D_ACN_1 | Defines track as PCM |
| audioTrackFormat | AT_00040103_01 | PCM_N3D_ACN_2 | Defines track as PCM |
| audioTrackFormat | AT_00040104_01 | PCM_N3D_ACN_3 | Defines track as PCM |
| audioStreamFormat | AS_00040101 | PCM_N3D_ACN_0 | Defines stream as PCM |
| audioStreamFormat | AS_00040102 | PCM_N3D_ACN_1 | Defines stream as PCM |
| audioStreamFormat | AS_00040103 | PCM_N3D_ACN_2 | Defines stream as PCM |
| audioStreamFormat | AS_00040104 | PCM_N3D_ACN_3 | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00040101<br>AB_00040101_00000001 | N3D_ACN_0 | Describes channel as ACN0 HOA component |
| audioChannelFormat and audioBlockFormat | AC_00040102<br>AB_00040102_00000001 | N3D_ACN_1 | Describes channel as ACN1 HOA component |
| audioChannelFormat and audioBlockFormat | AC_00040103<br>AB_00040103_00000001 | N3D_ACN_2 | Describes channel as ACN2 HOA component |
| audioChannelFormat and audioBlockFormat | AC_00040104<br>AB_00040104_00000001 | N3D_ACN_3 | Describes channel as ACN3 HOA component |
| audioPackFormat | AP_00040011 | 3D_order1_N3D_ACN | Defines a 1st order HOA pack referring to four ACN channels. |

These are the elements in the content part of the description:

TABLE 66

**Scene-based example content elements**

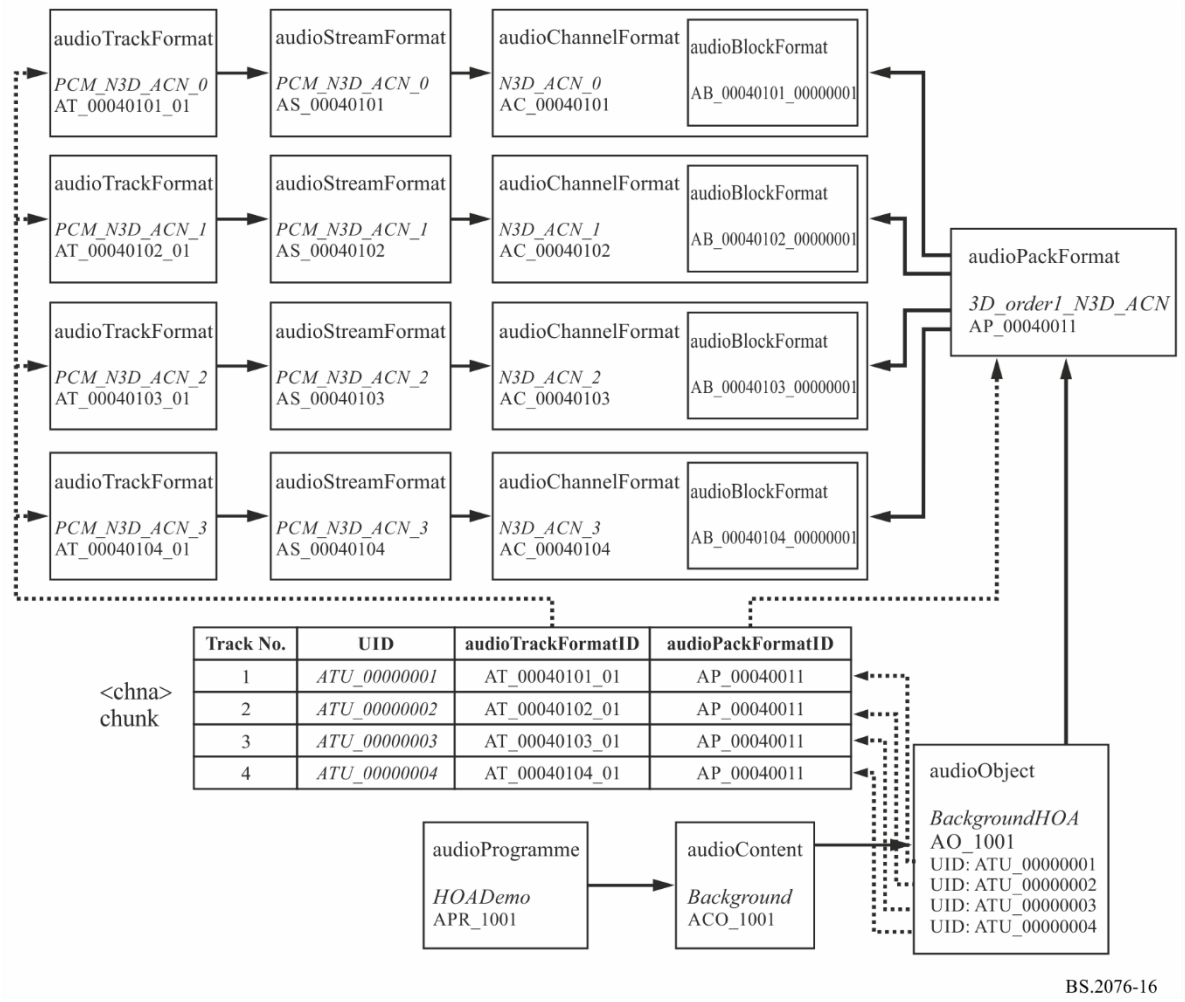
| Element | ID | Name | Description |
|---|---|---|---|
| audioObject | AO_1001 | BackgroundHOA | Object for 'BackgroundHOA', 1st order HOA format |
| audioContent | ACO_1001 | Background | 'Background' content |
| audioProgramme | APR_1001 | HOADemo | 'HOADemo' containing a 'Background'content |

## 3.2 Element Relationships

The diagram in Fig. 16 shows how the defined elements relate to each other. The top half of the diagram covers the elements that describe the 4 channels of the 1st order HOA (N3D method). The <*chna*> chunk in the middle shows how the four tracks are connected to the format definitions. The content definition elements are at the bottom of the diagram, with the audioObject element containing the track UID references to the UIDs in the <*chna*> chunk.

FIGURE 16

**Scene-based example diagram**



### 3.3 Sample code

This XML sample code does not include the audioFormatExtended parent element and the XML header for clarity. The first excerpt of code covers the format elements, which could be contained within the common reference file:

```
<!-- ############ -->
<!-- PACKS -->
<!-- ############ -->

<audioPackFormat audioPackFormatID="AP_00040011" audioPackFormatName="3D_order1_N3D_ACN"
typeLabel="0004"  typeDefinition="HOA">
  <normalization>N3D</normalization>
  <audioChannelFormatIDRef>AC_00040101</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00040102</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00040103</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00040104</audioChannelFormatIDRef>
</audioPackFormat>

<!-- ############ -->
<!-- CHANNELS -->
<!-- ############ -->

<audioChannelFormat audioChannelFormatID="AC_00040101"
audioChannelFormatName="N3D_ACN_0" typeDefinition="HOA">
  <audioBlockFormat audioBlockFormatID="AB_00040101_00000001">
    <degree>0</degree>
    <order>0</order>
    <normalization>N3D</normalization>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00040102"
audioChannelFormatName="N3D_ACN_1" typeDefinition="HOA">
  <audioBlockFormat audioBlockFormatID="AB_00040102_00000001">
    <degree>1</degree>
    <order>-1</order>
    <normalization>N3D</normalization>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00040103"
audioChannelFormatName="N3D_ACN_2" typeDefinition="HOA">
  <audioBlockFormat audioBlockFormatID="AB_00040103_00000001">
    <degree>1</degree>
    <order>0</order>
    <normalization>N3D</normalization>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00040104"
audioChannelFormatName="N3D_ACN_3" typeDefinition="HOA">
  <audioBlockFormat audioBlockFormatID="AB_00040104_00000001">
    <degree>1</degree>
    <order>1</order>
    <normalization>N3D</normalization>
  </audioBlockFormat>
</audioChannelFormat>
```

```
<!-- ############ -->
<!-- STREAMS -->
<!-- ############ -->

<audioStreamFormat audioStreamFormatID="AS_00040101"
audioStreamFormatName="PCM_N3D_ACN_0" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00040101</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00040101_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00040102"
audioStreamFormatName="PCM_N3D_ACN_1" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00040102</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00040102_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00040103"
audioStreamFormatName="PCM_N3D_ACN_2" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00040103</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00040103_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00040104"
audioStreamFormatName="PCM_N3D_ACN_3" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00040104</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00040104_01</audioTrackFormatIDRef>
</audioStreamFormat>

<!-- ############ -->
<!-- AUDIO TRACKS -->
<!-- ############ -->

<audioTrackFormat audioTrackFormatID="AT_00040101_01"
audioTrackFormatName="PCM_N3D_ACN_0" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00040101</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00040102_01"
audioTrackFormatName="PCM_N3D_ACN_1" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00040102</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00040103_01"
audioTrackFormatName="PCM_N3D_ACN_2" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00040103</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00040104_01"
audioTrackFormatName="PCM_N3D_ACN_3" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00040104</audioStreamFormatIDRef>
</audioTrackFormat>
```

The second excerpt covers the content part, which would have to be included in the <*axml*> chunk of the BWF file:

```
<!-- ############ -->
<!-- PROGRAMMES -->
<!-- ############ -->

<audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="HOADemo">
  <audioContentIDRef>ACO_1001</audioContentIDRef>
</audioProgramme>

<!-- ############ -->
<!-- CONTENTS -->
<!-- ############ -->

<audioContent audioContentID="ACO_1001" audioContentName="Background">
  <audioObjectIDRef>AO_1001</audioObjectIDRef>
</audioContent>

<!-- ############ -->
<!-- OBJECTS -->
<!-- ############ -->

<audioObject audioObjectID="AO_1001" audioObjectName="BackgroundHOA">
  <audioPackFormatIDRef>AP_00040011</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000003</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000004</audioTrackUIDRef>
</audioObject>

<!-- ############ -->
<!-- AUDIO TRACK UIDs -->
<!-- ############ -->

<audioTrackUID UID="ATU_00000001">
  <audioTrackFormatIDRef>AT_00040101_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00040011</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000002">
  <audioTrackFormatIDRef>AT_00040102_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00040011</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000003">
  <audioTrackFormatIDRef>AT_00040103_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00040011</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000004">
  <audioTrackFormatIDRef>AT_00040104_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00040011</audioPackFormatIDRef>
</audioTrackUID>
```

## 4 Material exchange format mapping example

The ADM has been designed to not only allow BW64 files to become a flexible multichannel file format, but also be incorporated by other file formats. Currently, material exchange format ((MXF) – SMPTE 377M), which carries both video and audio, has a rather limited capability in terms of specifying its audio format. The ADM could be used by MXF files in a similar way to BW64 files allowing a comprehensive format description of the audio.

MXF files often use EBU R123[1] ("EBU Audio Track Allocation for File Exchange") audio track configurations. This is a set of channel and matrix-based track allocations for between 2 and 16 track files or streams. This example will show how a particular R123 configuration can be represented by the ADM that is suitable for MXF.

This example will demonstrate how the 4a R123 configuration can be represented by the ADM. This configuration uses 4 tracks:

TABLE 67

**MXF example track configuration**

| Track Number | Track Use | Group |
|---|---|---|
| 1 | Stereo Left (PCM) | PCM Stereo pair |
| 2 | Stereo Right (PCM) | |
| 3 | MCA (coded audio) | Multichannel audio coded stream |
| 4 | MCA (coded audio) | |

### 4.1 Summary of elements

These are the elements in the format part of the description:

TABLE 68

**MXF example format elements**

| Element | ID | Name | Description |
|---|---|---|---|
| audioTrackFormat | AT_00010001_01 | PCM_FrontLeft | Defines track as PCM |
| audioTrackFormat | AT_00010002_01 | PCM_FrontRight | Defines track as PCM |
| audioTrackFormat | AT_10011001_01 | CodedAudio1 | Defines track as containing coded data |
| audioTrackFormat | AT_10011001_02 | CodedAudio2 | Defines track as containing coded data |
| audioStreamFormat | AS_00010001 | PCM_FrontLeft | Defines stream as PCM |
| audioStreamFormat | AS_00010002 | PCM_FrontRight | Defines stream as PCM |
| audioStreamFormat | AS_10011001 | CodedAudio_5.1 | Defines stream as coded data |
| audioChannelFormat & audioBlockFormat | AC_00010001 AB_00010001_00000001 | FrontLeft | Describes channel as front left with a position and speaker reference |
| audioChannelFormat & audioBlockFormat | AC_00010002 AB_00010002_00000001 | FrontRight | Describes channel as front right with a position and speaker reference |

[1] EBU R 123 – EBU Audio Track Allocation for File Exchange.

TABLE 68 (*end*)

| Element | ID | Name | Description |
|---|---|---|---|
| audioChannelFormat & audioBlockFormat | AC_00010003 AB_00010003_00000001 | FrontCentre | Describes channel as front centre with a position and speaker reference |
| audioChannelFormat & audioBlockFormat | AC_00010004 AB_00010004_00000001 | LFE | Describes channel as LFE with a position and speaker reference |
| audioChannelFormat & audioBlockFormat | AC_00010005 AB_00010005_00000001 | SurroundLeft | Describes channel as front right with a position and speaker reference |
| audioChannelFormat & audioBlockFormat | AC_00010006 AB_00010006_00000001 | SurroundRight | Describes channel as front right with a position and speaker reference |
| audioPackFormat | AP_00010002 | Stereo | Defines a stereo pack referring to two channels |
| audioPackFormat | AP_00010003 | 5.1 | Defines a 5.1 pack referring to six channels |

These are the elements in the content part of the description:

TABLE 69

**MXF example content elements**

| Element | ID | Name | Description |
|---|---|---|---|
| audioObject | AO_1041 | R123_4a | Object for R123 4a configuration |
| audioObject | AO_1002 | R123_Stereo | Object for stereo |
| audioObject | AO_1004 | R123_5.1 | Object for 5.1 |

## 4.2 Element Relationships

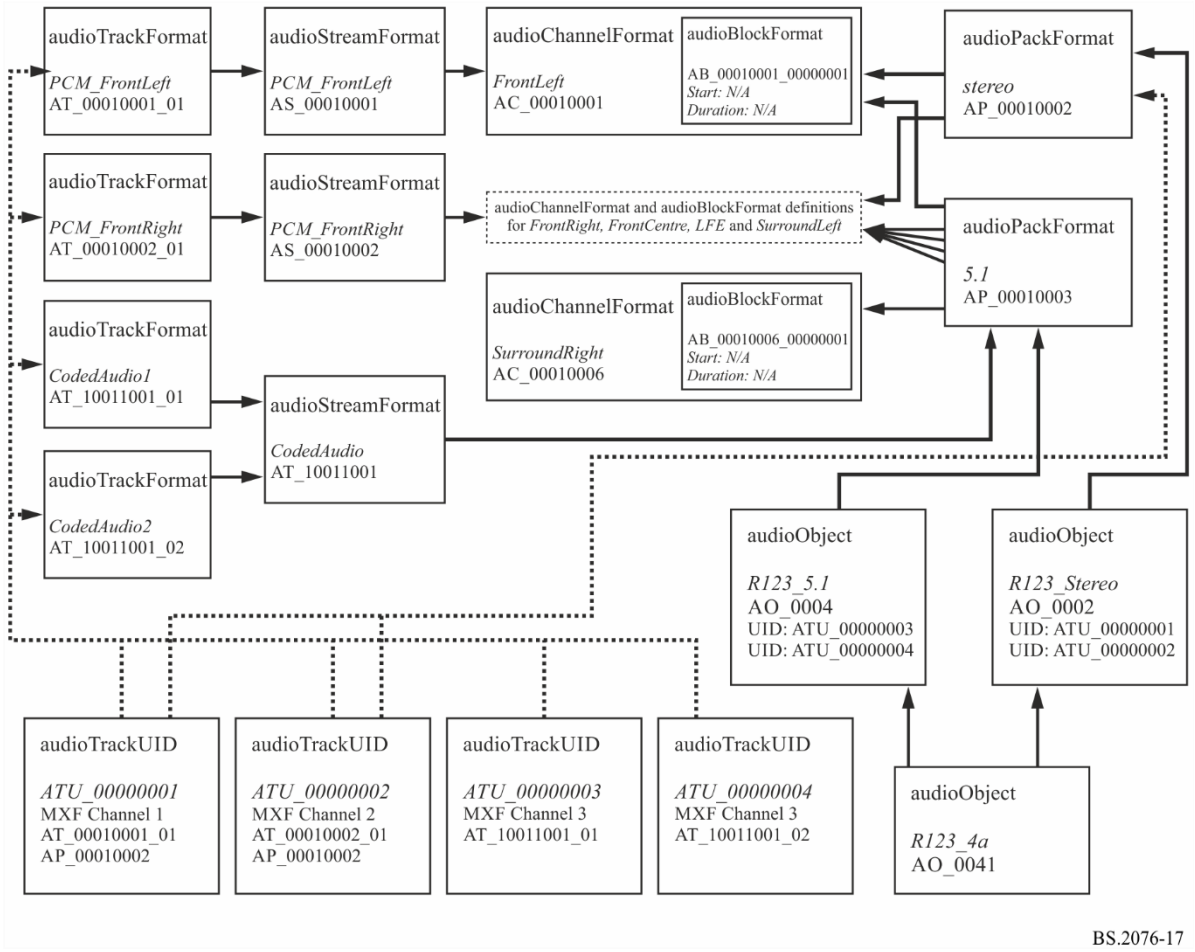
The diagram in Fig. 17 shows how the defined elements relate to each other. The top half of the diagram covers the elements that describe the two-channel stereo PCM format and the six-channel coded audio 5.1 encoded format. In the coded audio part, two audioTrackFormats refer to a single audioStreamFormat as coded audio requires that two tracks are combined to decode the audio signals. The coded audio audioStreamFormat refers to an audioPackFormat as it is representing a group of channels rather than a single one. This 5.1 audioPackFormat refers to the six audioChannelFormats that describe each channel.

The R123 4a configuration is represented by an audioObject (named ‘R123_4a’), which refers to two further audioObjects (for the stereo and 5.1 groups), which contain the references to the audioTrackUIDs. This demonstrates the nesting feature of audioObjects.

As MXF does not feature a <*chna*> chunk, it uses sub-elements of audioTrackUID to generate references to the essences within the MXF file. The audioMXFLookUp sub-element is designed to facilitate these relationships.

FIGURE 17

**MXF mapping example diagram**




## 4.3 Sample code

This XML sample code does not include the audioFormatExtended parent element and the XML header for clarity. The first excerpt of code covers the format elements, which could be contained within the common reference file:

```
<!-- ############ -->
<!-- PACKS -->
<!-- ############ -->

<audioPackFormat audioPackFormatID="AP_00010002" audioPackFormatName="Stereo"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
</audioPackFormat>

<audioPackFormat audioPackFormatID="AP_00010003" audioPackFormatName="5.1"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010003</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010004</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010005</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010006</audioChannelFormatIDRef>
</audioPackFormat>
```

```
<!-- ############ -->
<!-- CHANNELS -->
<!-- ############ -->

<audioChannelFormat audioChannelFormatID="AC_00010001"
audioChannelFormatName="FrontLeft" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010001_00000001">
    <speakerLabel>M+030</speakerLabel>
    <position coordinate="azimuth">30.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010002"
audioChannelFormatName="FrontRight" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010002_00000001">
    <speakerLabel>M-030</speakerLabel>
    <position coordinate="azimuth">-30.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010003"
audioChannelFormatName="FrontCentre" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010003_00000001">
    <speakerLabel>M+000</speakerLabel>
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010004" audioChannelFormatName="LFE"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <frequency typeDefinition="lowPass">120</frequency>
  <audioBlockFormat audioBlockFormatID="AB_00010004_00000001">
    <speakerLabel>LFE</speakerLabel>
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">-20.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010005"
audioChannelFormatName="SurroundLeft" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010005_00000001">
    <speakerLabel>M+110</speakerLabel>
    <position coordinate="azimuth">110.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010006"
audioChannelFormatName="SurroundRight" typeLabel="0001"
typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010006_00000001">
    <speakerLabel>M-110</speakerLabel>
```

```
    <position coordinate="azimuth">-110.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<!-- ############ -->
<!-- STREAMS -->
<!-- ############ -->

<audioStreamFormat audioStreamFormatID="AS_00010001"
audioStreamFormatName="PCM_FrontLeft" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010002"
audioStreamFormatName="PCM_FrontRight" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_10011001"
audioStreamFormatName="CodedAudio_5.1" formatLabel="1001"
formatDefinition="CodedAudio">
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
  <audioTrackFormatIDRef>AT_10011001_01</audioTrackFormatIDRef>
  <audioTrackFormatIDRef>AT_10011001_02</audioTrackFormatIDRef>
</audioStreamFormat>

<!-- ############ -->
<!-- AUDIO TRACKS -->
<!-- ############ -->
<audioTrackFormat audioTrackFormatID="AT_00010001_01"
audioTrackFormatName="PCM_FrontLeft" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010001</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010002_01"
audioTrackFormatName="PCM_FrontRight" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010002</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_10011001_01"
audioTrackFormatName="CodedAudio1" formatLabel="1001" formatDefinition="data">
  <audioStreamFormatIDRef>AS_10011001</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_10011001_02"
audioTrackFormatName="CodedAudio2" formatLabel="1001" formatDefinition="data">
  <audioStreamFormatIDRef>AS_10011001</audioStreamFormatIDRef>
</audioTrackFormat>
```

The second excerpt (below) covers the content part, in this case audioObjects and audioTrackUIDs, which should be contained within the MXF file. The audioTrackUIDs contain the audioMXFLoopUp elements that locate the essence within the MXF file.

```
<!-- ############ -->
<!-- OBJECTS -->
<!-- ############ -->
```

```
<audioObject audioObjectID="AO_1041" audioObjectName="R123_4a">
  <audioObjectIDRef>AO_1002</audioObjectIDRef>
  <audioObjectIDRef>AO_1004</audioObjectIDRef>
</audioObject>

<audioObject audioObjectID="AO_1002" audioObjectName="R123_Stereo">
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
</audioObject>

<audioObject audioObjectID="AO_1004" audioObjectName="R123_5.1coded">
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000003</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000004</audioTrackUIDRef>
</audioObject>

<!-- ############ -->
<!-- AUDIO TRACK UIDs -->
<!-- ############ -->

<audioTrackUID UID="ATU_00000001">
  <audioMXFLookUp>
    <packageUIDRef>urn:smpte:umid:060a2b34.01010105.01010f20.13000000.540bca53.41434f05.
8ce5f4e3.5b72c985</packageUIDRef>
    <trackIDRef>MXFTRACK_3</trackIDRef>
    <channelIDRef>MXFCHAN_1</channelIDRef>
  </audioMXFLookUp>
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000002">
  <audioMXFLookUp>
    <packageUIDRef>urn:smpte:umid:060a2b34.01010105.01010f20.13000000.540bca53.41434f05.
8ce5f4e3.5b72c985</packageUIDRef>
    <trackIDRef>MXFTRACK_3</trackIDRef>
    <channelIDRef>MXFCHAN_2</channelIDRef>
  </audioMXFLookUp>
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010002</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000003">
  <audioMXFLookUp>
    <packageUIDRef>urn:smpte:umid:060a2b34.01010105.01010f20.13000000.540bca53.41434f05.
8ce5f4e3.5b72c985</packageUIDRef>
    <trackIDRef>MXFTRACK_3</trackIDRef>
    <channelIDRef>MXFCHAN_1</channelIDRef>
  </audioMXFLookUp>
  <audioTrackFormatIDRef>AT_10011001_01</audioTrackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000004">
  <audioMXFLookUp>
    <packageUIDRef>urn:smpte:umid:060a2b34.01010105.01010f20.13000000.540bca53.41434f05.
8ce5f4e3.5b72c985</packageUIDRef>
    <trackIDRef>MXFTRACK_3</trackIDRef>
    <channelIDRef>MXFCHAN_1</channelIDRef>
  </audioMXFLookUp>
  <audioTrackFormatIDRef>AT_10011001_02</audioTrackFormatIDRef>
</audioTrackUID>
```

## 5 Personalized audio example

To demonstrate how the ADM can be used to describe personalized audio, here is an example using a combination of channel-based audio for the ambience/bed and object-based audio for the commentator objects. This example uses multiple audioProgramme elements that represent five different preset mixes for a sports programme: default mix, just the action, clear commentary, home team, and away team. The corresponding ADM XML tree contains four different audioContent elements to choose from: ambience, main commentary, home team biased commentary, and away team biased commentary.

TABLE 70

**Personalized audio example mixes**

| | **Ambience** | **Main commentary 1** | **Main commentary 2** | **Home team biased commentary** | **Away team biased commentary** |
|---|---|---|---|---|---|
| Default mix | • | • | • | | |
| Just the action | • | | | | |
| Clear commentary | | • | • | | |
| Home team | • | | | • | |
| Away team | • | | | | • |

### 5.1 Summary of elements

These are the elements in the format part of the description:

TABLE 71

**Personalized example format elements**

| **Element** | **ID** | **Name** | **Description** |
|---|---|---|---|
| audioTrackFormat | AT_00010001_01 | PCM_FrontLeft | Defines track as PCM |
| audioStreamFormat | AS_00010001 | PCM_FrontLeft | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010001<br>AB_00010001_00000001 | FrontLeft | Describes channel as front left with a position and speaker reference |
| audioTrackFormat | AT_00010002_01 | PCM_FrontRight | Defines track as PCM |
| audioStreamFormat | AS_00010002 | PCM_FrontRight | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010002<br>AB_00010002_00000001 | FrontRight | Describes channel as front right with a position and speaker reference |
| audioTrackFormat | AT_00010003_01 | PCM_FrontCentre | Defines track as PCM |
| audioStreamFormat | AS_00010003 | PCM_FrontCentre | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010003<br>AB_00010003_00000001 | FrontCentre | Describes channel as front centre with a position and speaker reference |

TABLE 71 (*end*)

| Element | ID | Name | Description |
|---|---|---|---|
| audioTrackFormat | AT_00010004_01 | PCM_LFE | Defines track as PCM |
| audioStreamFormat | AS_00010004 | PCM_LFE | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010004<br><br>AB_00010004_00000001 | LFE | Describes channel as LFE with a position and speaker reference |
| audioTrackFormat | AT_00010005_01 | PCM_SurroundLeft | Defines track as PCM |
| audioStreamFormat | AS_00010005 | PCM_SurroundLeft | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010005<br><br>AB_00010005_00000001 | SurroundLeft | Describes channel as surround left with a position and speaker reference |
| audioTrackFormat | AT_00010006_01 | PCM_SurroundRight | Defines track as PCM |
| audioStreamFormat | AS_00010006 | PCM_SurroundRight | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010006<br><br>AB_00010006_00000001 | SurroundRight | Describes channel as surround right with a position and speaker reference |
| audioPackFormat | AP_00010003 | 5.1 | Defines a 5.1 pack referring to six channels. |
| audioTrackFormat | AT_00031001_01 | PCM_Main_Comm1 | Defines track as PCM |
| audioStreamFormat | AS_00031001 | PCM_Main_Comm1 | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00031001<br><br>AB_00031001_00000001 | Main_Comm1 | Describes channel as an object type containing a single block with positional metadata. |
| audioTrackFormat | AT_00031002_01 | PCM_Main_Comm2 | Defines track as PCM |
| audioStreamFormat | AS_00031002 | PCM_Main_Comm2 | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00031002<br><br>AB_00031002_00000001 | Main_Comm2 | Describes channel as an object type containing a single block with positional metadata. |
| audioTrackFormat | AT_00031003_01 | PCM_Home_Comm | Defines track as PCM |
| audioStreamFormat | AS_00031003 | PCM_Home_Comm | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00031003<br><br>AB_00031003_00000001 | Home_Comm | Describes channel as an object type containing a single block with positional metadata. |
| audioTrackFormat | AT_00031004_01 | PCM_Away_Comm | Defines track as PCM |
| audioStreamFormat | AS_00031004 | PCM_Away_Comm | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00031004<br><br>AB_00031004_00000001 | Away_Comm | Describes channel as an object type containing a single block with positional metadata. |
| audioPackFormat | AP_00031001 | MainComm1 | Defines a pack referring to one channel. |
| audioPackFormat | AP_00031002 | MainComm2 | Defines a pack referring to one channel. |
| audioPackFormat | AP_00031003 | HomeComm | Defines a pack referring to one channel. |
| audioPackFormat | AP_00031004 | AwayComm | Defines a pack referring to one channel. |

TABLE 72

**Personlized example content elements**

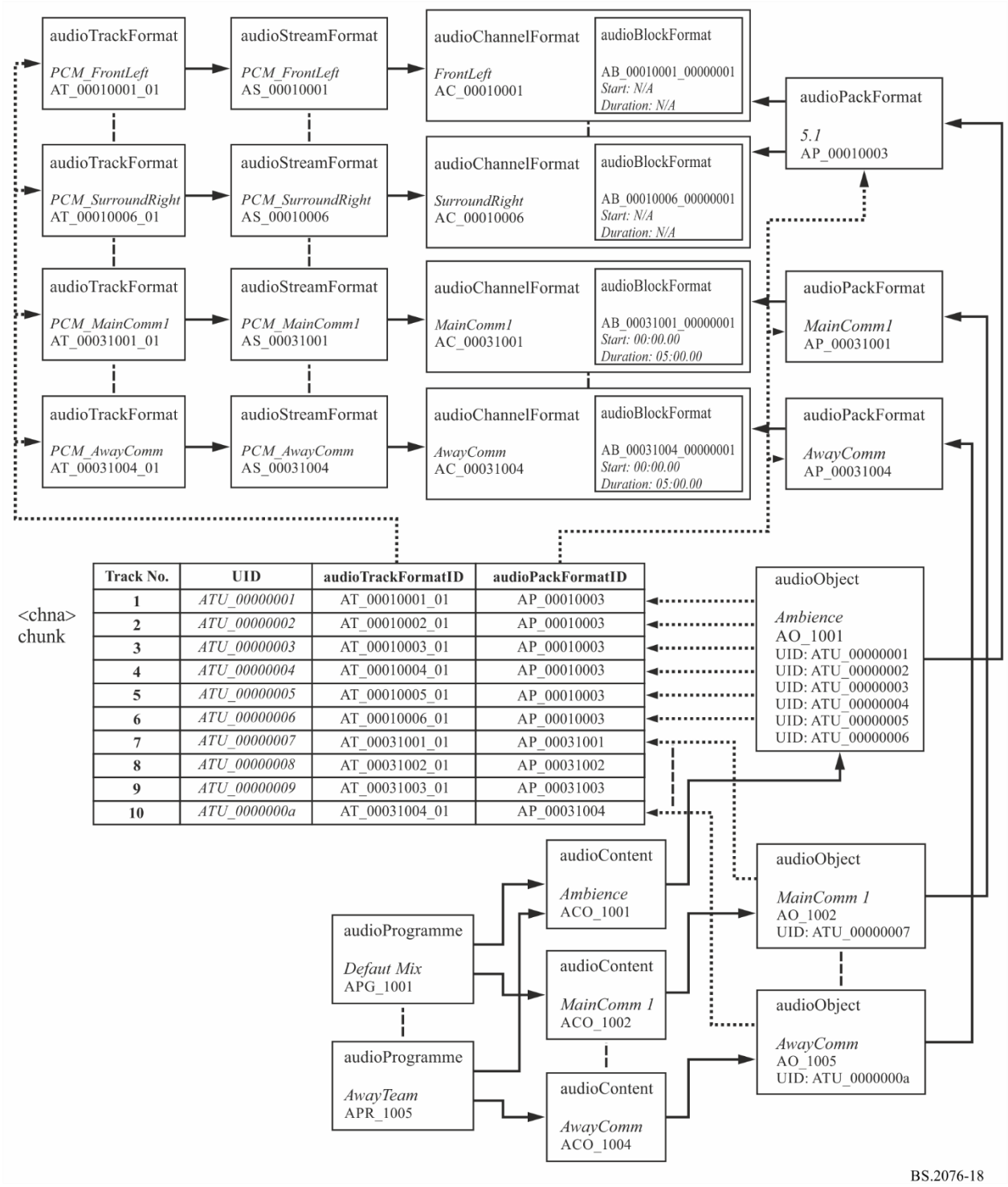
| Element | ID | Name | Description |
|---|---|---|---|
| audioObject | AO_1001 | Ambience | Object for ‘Ambience’, 5.1 format |
| audioContent | ACO_1001 | Ambience | ‘Ambience’ content |
| audioObject | AO_1002 | Main_Comm1 | Object for ‘Main_Comm1’, mono format |
| audioObject | AO_1003 | Main_Comm2 | Object for ‘Main_Comm2’, mono format |
| audioContent | ACO_1002 | Main_Comm | ‘Main_Comm’ content |
| audioObject | AO_1004 | Home_Comm | Object for ‘Home_Comm’, mono format |
| audioContent | ACO_1003 | Home_Comm | ‘Home_Comm’ content |
| audioObject | AO_1005 | Away_Comm | Object for ‘Away  Comm’, mono format |
| audioContent | ACO_1004 | Away_Comm | ‘Away_Comm’ content |
| audioProgramme | APR_1001 | DefaultMix | Programme ‘DefaultMix’ containing ‘Ambience’ and ‘Main_Comm” content |
| audioProgramme | APR_1002 | JustTheAction | Programme ‘JustTheAction’ containing only ‘Ambience’ content |
| audioProgramme | APR_1003 | ClearCommentary | Programme ‘ClearCommentary’ containing only ‘Main_Comm’ content |
| audioProgramme | APR_1004 | HomeTeam | Programme ‘HomeTeam’ containing ‘Ambience’ and ‘Home_Comm’ content |
| audioProgramme | APR_1005 | AwayTeam | Programme ‘AwayTeam’ containing ‘Ambience’ and ‘Away_Comm’ content |

## 5.2 Element Relationships

The diagram in Fig. 18 shows how the defined elements relate to each other. The top half of the diagram covers the elements that describe the 5.1 channel ambience/bed and the 4 mono objects. The <*chna*> chunk in the middle shows how the tracks are connected to the format definitions. The content definition elements are at the bottom of the diagram, with the audioObject element containing the track UID references to the UID in the <*chna*> chunk.

FIGURE 18

**Personalised audio example diagram**




## 5.3 Sample code

This XML sample code does not include the audioFormatExtended parent element and the XML header for clarity. The excerpt of code covers both the format and content elements:

```
<!-- ############ -->
<!-- PROGRAMMES -->
<!-- ############ -->

<audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="DefaultMix">
```

```
  <audioContentIDRef>ACO_1001</audioContentIDRef>
  <audioContentIDRef>ACO_1002</audioContentIDRef>
</audioProgramme>

<audioProgramme audioProgrammeID="APR_1002" audioProgrammeName="JustTheAction">
  <audioContentIDRef>ACO_1001</audioContentIDRef>
</audioProgramme>

<audioProgramme audioProgrammeID="APR_1003" audioProgrammeName="ClearCommentary">
  <audioContentIDRef>ACO_1002</audioContentIDRef>
</audioProgramme>

<audioProgramme audioProgrammeID="APR_1004" audioProgrammeName="HomeTeam">
  <audioContentIDRef>ACO_1001</audioContentIDRef>
  <audioContentIDRef>ACO_1003</audioContentIDRef>
</audioProgramme>

<audioProgramme audioProgrammeID="APR_1005" audioProgrammeName="AwayTeam">
  <audioContentIDRef>ACO_1001</audioContentIDRef>
  <audioContentIDRef>ACO_1004</audioContentIDRef>
</audioProgramme>

<!-- ############ -->
<!-- CONTENTS -->
<!-- ############ -->

<audioContent audioContentID="ACO_1001" audioContentName="Ambience">
  <audioObjectIDRef>AO_1001</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-23.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<audioContent audioContentID="ACO_1002" audioContentName="Main_Comm">
  <audioObjectIDRef>AO_1002</audioObjectIDRef>
  <audioObjectIDRef>AO_1003</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-23.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<audioContent audioContentID="ACO_1003" audioContentName="Home_Comm">
  <audioObjectIDRef>AO_1004</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-23.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<audioContent audioContentID="ACO_1004" audioContentName="AwayComm">
  <audioObjectIDRef>AO_1005</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-23.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<!-- ############ -->
<!-- OBJECTS -->
<!-- ############ -->
```

```
<audioObject audioObjectID="AO_1001" audioObjectName="Ambience">
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000003</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000004</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000005</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000006</audioTrackUIDRef>
</audioObject>

<audioObject audioObjectID="AO_1002" audioObjectName="Main_Comm1"
start="00:00:00.00000">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000007</audioTrackUIDRef>
</audioObject>

<audioObject audioObjectID="AO_1003" audioObjectName="Main_Comm2"
start="00:00:00.00000">
  <audioPackFormatIDRef>AP_00031002</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000008</audioTrackUIDRef>
</audioObject>

<audioObject audioObjectID="AO_1004" audioObjectName="Home_Comm"
start="00:00:00.00000">
  <audioPackFormatIDRef>AP_00031003</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000009</audioTrackUIDRef>
</audioObject>

<audioObject audioObjectID="AO_1005" audioObjectName="Away_Comm"
start="00:00:00.00000">
  <audioPackFormatIDRef>AP_00031004</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_0000000a</audioTrackUIDRef>
</audioObject>

<!-- ############ -->
<!-- PACKS -->
<!-- ############ -->

<audioPackFormat audioPackFormatID="AP_00010003" audioPackFormatName="5.1"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010003</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010004</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010005</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010006</audioChannelFormatIDRef>
</audioPackFormat>

<audioPackFormat audioPackFormatID="AP_00031001" audioPackFormatName="MainComm1"
typeLabel="0003" typeDefinition="Objects">
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
</audioPackFormat>

<audioPackFormat audioPackFormatID="AP_00031002" audioPackFormatName="MainComm2"
typeLabel="0003" typeDefinition="Objects">
  <audioChannelFormatIDRef>AC_00031002</audioChannelFormatIDRef>
</audioPackFormat>
```

```
<audioPackFormat audioPackFormatID="AP_00031003" audioPackFormatName="HomeComm"
typeLabel="0003" typeDefinition="Objects">
  <audioChannelFormatIDRef>AC_00031003</audioChannelFormatIDRef>
</audioPackFormat>

<audioPackFormat audioPackFormatID="AP_00031004" audioPackFormatName="AwayComm"
typeLabel="0003" typeDefinition="Objects">
  <audioChannelFormatIDRef>AC_00031004</audioChannelFormatIDRef>
</audioPackFormat>

<!-- ############ -->
<!-- CHANNELS -->
<!-- ############ -->

<audioChannelFormat audioChannelFormatID="AC_00010001"
audioChannelFormatName="FrontLeft" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010001_00000001">
    <speakerLabel>M+030</speakerLabel>
    <position coordinate="azimuth">30.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010002"
audioChannelFormatName="FrontRight" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010002_00000001">
    <speakerLabel>M-030</speakerLabel>
    <position coordinate="azimuth">-30.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010003"
audioChannelFormatName="FrontCentre" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010003_00000001">
    <speakerLabel>M+000</speakerLabel>
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010004" audioChannelFormatName="LFE"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <frequency typeDefinition="lowPass">200</frequency>
  <audioBlockFormat audioBlockFormatID="AB_00010004_00000001">
    <speakerLabel>LFE</speakerLabel>
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">-20.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010005"
audioChannelFormatName="SurroundLeft" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010005_00000001">
    <speakerLabel>M+110</speakerLabel>
```

```
    <position coordinate="azimuth">110.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010006"
audioChannelFormatName="SurroundRight" typeLabel="0001"
typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010006_00000001">
    <speakerLabel>M-110</speakerLabel>
    <position coordinate="azimuth">-110.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00031001"
audioChannelFormatName="MainComm1" typeLabel="0003" typeDefinition="Objects">
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
duration="00:05:00.00000">
    <position coordinate="X">-1.0</position>
    <position coordinate="Y">1.0</position>
    <position coordinate="Z">0.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00031002"
audioChannelFormatName="MainComm2" typeLabel="0003" typeDefinition="Objects">
  <audioBlockFormat audioBlockFormatID="AB_00031002_00000001" rtime="00:00:00.00000"
duration="00:05:00.00000">
    <position coordinate="X">1.0</position>
    <position coordinate="Y">1.0</position>
    <position coordinate="Z">0.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00031003"
audioChannelFormatName="HomeComm" typeLabel="0003" typeDefinition="Objects">
  <audioBlockFormat audioBlockFormatID="AB_00031003_00000001" rtime="00:00:00.00000"
duration="00:05:00.00000">
    <position coordinate="X">0.0</position>
    <position coordinate="Y">1.0</position>
    <position coordinate="Z">0.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00031004"
audioChannelFormatName="AwayComm" typeLabel="0003" typeDefinition="Objects">
  <audioBlockFormat audioBlockFormatID="AB_00031004_00000001" rtime="00:00:00.00000"
duration="00:05:00.00000">
    <position coordinate="X">0.0</position>
    <position coordinate="Y">1.0</position>
    <position coordinate="Z">0.0</position>
  </audioBlockFormat>
</audioChannelFormat>


<!-- ############ -->
<!-- STREAMS -->
```

```
<!-- ############ -->

<audioStreamFormat audioStreamFormatID="AS_00010001"
audioStreamFormatName="PCM_FrontLeft" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010002"
audioStreamFormatName="PCM_FrontRight" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010003"
audioStreamFormatName="PCM_FrontCentre" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010003</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010003_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010004" audioStreamFormatName="PCM_LFE"
formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010004</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010004_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010005"
audioStreamFormatName="PCM_SurroundLeft" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010005</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010005_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010006"
audioStreamFormatName="PCM_SurroundRight" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010006</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010006_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00031001"
audioStreamFormatName="PCM_MainComm1" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00031002"
audioStreamFormatName="PCM_MainComm2" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00031002</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00031002_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00031003"
audioStreamFormatName="PCM_HomeComm" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00031003</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00031003_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00031004"
audioStreamFormatName="PCM_AwayComm" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00031004</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00031004_01</audioTrackFormatIDRef>
```

```
</audioStreamFormat>

<!-- ############ -->
<!-- AUDIO TRACKS -->
<!-- ############ -->

<audioTrackFormat audioTrackFormatID="AT_00010001_01"
audioTrackFormatName="PCM_FrontLeft" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010001</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010002_01"
audioTrackFormatName="PCM_FrontRight" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010002</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010003_01"
audioTrackFormatName="PCM_FrontCentre" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010003</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010004_01" audioTrackFormatName="PCM_LFE"
formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010004</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010005_01"
audioTrackFormatName="PCM_SurroundLeft" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010005</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010006_01"
audioTrackFormatName="PCM_SurroundRight" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010006</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00031001_01"
audioTrackFormatName="PCM_MainComm1" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00031001</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00031002_01"
audioTrackFormatName="PCM_MainComm2" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00031002</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00031003_01"
audioTrackFormatName="PCM_HomeComm" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00031003</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00031004_01"
audioTrackFormatName="PCM_AwayComm" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00031004</audioStreamFormatIDRef>
</audioTrackFormat>

<!-- ############ -->
<!-- AUDIO TRACK UIDs -->
<!-- ############ -->
```

```
<audioTrackUID UID="ATU_00000001">
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000002">
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000003">
  <audioTrackFormatIDRef>AT_00010003_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000004">
  <audioTrackFormatIDRef>AT_00010004_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000005">
  <audioTrackFormatIDRef>AT_00010005_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000006">
  <audioTrackFormatIDRef>AT_00010006_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010003</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000007">
  <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000008">
  <audioTrackFormatIDRef>AT_00031002_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00031002</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000009">
  <audioTrackFormatIDRef>AT_00031003_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00031003</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_0000000a">
  <audioTrackFormatIDRef>AT_00031004_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00031004</audioPackFormatIDRef>
</audioTrackUID>
```

# 6 22.2 multichannel programme with an alternative dialogue example

## 6.1 Summary of elements

These are the elements in the format part of the description:

TABLE 73

**22.2 example format elements**

| Element | ID | Name | Description |
|---|---|---|---|
| audioTrackFormat | AT_00010018_01 | PCM_FrontLeftWide | Defines track as PCM |
| audioStreamFormat | AS_00010018 | PCM_FrontLeftWide | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010018<br>AB_00010018_00000001 | FrontLeftWide | Describes channel as front left with a position and speaker reference |
| audioTrackFormat | AT_00010019_01 | PCM_FrontRightWide | Defines track as PCM |
| audioStreamFormat | AS_00010019 | PCM_FrontRightWide | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010019<br>AB_00010019_00000001 | FrontRightWide | Describes channel as front right with a position and speaker reference |
| audioTrackFormat | AT_00010003_01 | PCM_FrontCentre | Defines track as PCM |
| audioStreamFormat | AS_00010003 | PCM_FrontCentre | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010003<br>AB_00010003_00000001 | FrontCentre | Describes channel as front centre with a position and speaker reference |
| audioTrackFormat | AT_00010020_01 | PCM_LFE1 | Defines track as PCM |
| audioStreamFormat | AS_00010020 | PCM_LFE1 | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010020<br>AB_00010020_00000001 | LFE1 | Describes channel as LFE1 with a position and speaker reference |
| audioTrackFormat | AT_0001001c_01 | PCM_BackLeftMid | Defines track as PCM |
| audioStreamFormat | AS_0001001c | PCM_BackLeftMid | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_0001001c<br>AB_0001001c_00000001 | BackLeftMid | Describes channel as surround left with a position and speaker reference |
| audioTrackFormat | AT_0001001d_01 | PCM_BackRightMid | Defines track as PCM |
| audioStreamFormat | AS_0001001d | PCM_BackRightMid | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_0001001d<br>AB_0001001d_00000001 | BackRightMid | Describes channel as surround right with a position and speaker reference |
| audioTrackFormat | AT_00010001_01 | PCM_FrontLeft | Defines track as PCM |
| audioStreamFormat | AS_00010001 | PCM_FrontLeft | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010001<br>AB_00010001_00000001 | FrontLeft | Describes channel as front left centre with a position and speaker reference |
| audioTrackFormat | AT_00010002_01 | PCM_FrontRight | Defines track as PCM |
| audioStreamFormat | AS_00010002 | PCM_FrontRight | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010002<br>AB_00010002_00000001 | FrontRight | Describes channel as front right centre with a position and speaker reference |
| audioTrackFormat | AT_00010009_01 | PCM_BackCentre | Defines track as PCM |
| audioStreamFormat | AS_00010009 | PCM_BackCentre | Defines stream as PCM |

TABLE 73 (*continued*)

| Element | ID | Name | Description |
|---|---|---|---|
| audioChannelFormat and audioBlockFormat | AC_00010009<br>AB_00010009_00000001 | BackCentre | Describes channel as back centre with a position and speaker reference |
| audioTrackFormat | AT_00010021_01 | PCM_LFE2 | Defines track as PCM |
| audioStreamFormat | AS_00010021 | PCM_LFE2 | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010021<br>AB_00010021_00000001 | LFE2 | Describes channel as LFE2 with a position and speaker reference |
| audioTrackFormat | AT_0001000a_01 | PCM_SideLeft | Defines track as PCM |
| audioStreamFormat | AS_0001000a | PCM_SideLeft | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_0001000a<br>AB_0001000a_00000001 | SideLeft | Describes channel as surround right with a position and speaker reference |
| audioTrackFormat | AT_0001000b_01 | PCM_SideRight | Defines track as PCM |
| audioStreamFormat | AS_0001000b | PCM_SideRight | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_0001000b<br>AB_0001000b_00000001 | SideRight | Describes channel as side right with a position and speaker reference |
| audioTrackFormat | AT_00010022_01 | PCM_TopFrontLeftMid | Defines track as PCM |
| audioStreamFormat | AS_00010022 | PCM_TopFrontLeftMid | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010022<br>AB_00010022_00000001 | TopFrontLeftMid | Describes channel as top front left with a position and speaker reference |
| audioTrackFormat | AT_00010023_01 | PCM_TopFrontRightMid | Defines track as PCM |
| audioStreamFormat | AS_00010023 | PCM_TopFrontRightMid | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010023<br>AB_00010023_00000001 | TopFrontRightMid | Describes channel as top front right with a position and speaker reference |
| audioTrackFormat | AT_0001000e_01 | PCM_TopFrontCentre | Defines track as PCM |
| audioStreamFormat | AS_0001000e | PCM_TopFrontCentre | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_0001000e<br>AB_0001000e_00000001 | TopFrontCentre | Describes channel as top front centre with a position and speaker reference |
| audioTrackFormat | AT_0001000c_01 | PCM_TopCentre | Defines track as PCM |
| audioStreamFormat | AS_0001000c | PCM_TopCentre | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_0001000c<br>AB_0001000c_00000001 | TopCentre | Describes channel as top centre with a position and speaker reference |
| audioTrackFormat | AT_0001001e_01 | PCM_TopBackLeftMid | Defines track as PCM |
| audioStreamFormat | AS_0001001e | PCM_TopBackLeftMid | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_0001001e<br>AB_0001001e_00000001 | TopBackLeftMid | Describes channel as top back left with a position and speaker reference |
| audioTrackFormat | AT_0001001f_01 | PCM_TopBackRightMid | Defines track as PCM |
| audioStreamFormat | AS_0001001f | PCM_TopBackRightMid | Defines stream as PCM |

TABLE 73 (*end*)

| Element | ID | Name | Description |
|---|---|---|---|
| audioChannelFormat and audioBlockFormat | AC_0001001f<br>AB_0001001f_00000001 | TopBackRightMid | Describes channel as top back right with a position and speaker reference |
| audioTrackFormat | AT_00010013_01 | PCM_TopSideLeft | Defines track as PCM |
| audioStreamFormat | AS_00010013 | PCM_TopSideLeft | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010013<br>AB_00010013_00000001 | TopSideLeft | Describes channel as top side left with a position and speaker reference |
| audioTrackFormat | AT_00010014_01 | PCM_TopSideRight | Defines track as PCM |
| audioStreamFormat | AS_00010014 | PCM_TopSideRight | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010014<br>AB_00010014_00000001 | TopSideRight | Describes channel as top side right with a position and speaker reference |
| audioTrackFormat | AT_00010011_01 | PCM_TopBackCentre | Defines track as PCM |
| audioStreamFormat | AS_00010011 | PCM_TopBackCentre | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010011<br>AB_00010011_00000001 | TopBackCentre | Describes channel as top back centre with a position and speaker reference |
| audioTrackFormat | AT_00010015_01 | PCM_BottomFrontCentre | Defines track as PCM |
| audioStreamFormat | AS_00010015 | PCM_BottomFrontCentre | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010015<br>AB_00010015_00000001 | BottomFrontCentre | Describes channel as bottom front centre with a position and speaker reference |
| audioTrackFormat | AT_00010016_01 | PCM_BottomFrontLeftMid | Defines track as PCM |
| audioStreamFormat | AS_00010016 | PCM_BottomFrontLeftMid | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010016<br>AB_00010016_00000001 | BottomFrontLeftMid | Describes channel as bottom front left with a position and speaker reference |
| audioTrackFormat | AT_00010017_01 | PCM_BottomFrontRightMid | Defines track as PCM |
| audioStreamFormat | AS_00010017 | PCM_BottomFrontRightMid | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00010017<br>AB_00010017_00000001 | BottomFrontRightMid | Describes channel as bottom front right with a position and speaker reference |
| audioPackFormat | AP_00010009 | 22.2 | Defines a 22.2 pack referring to 24 channels. |

TABLE 74

**22.2 example content elements**

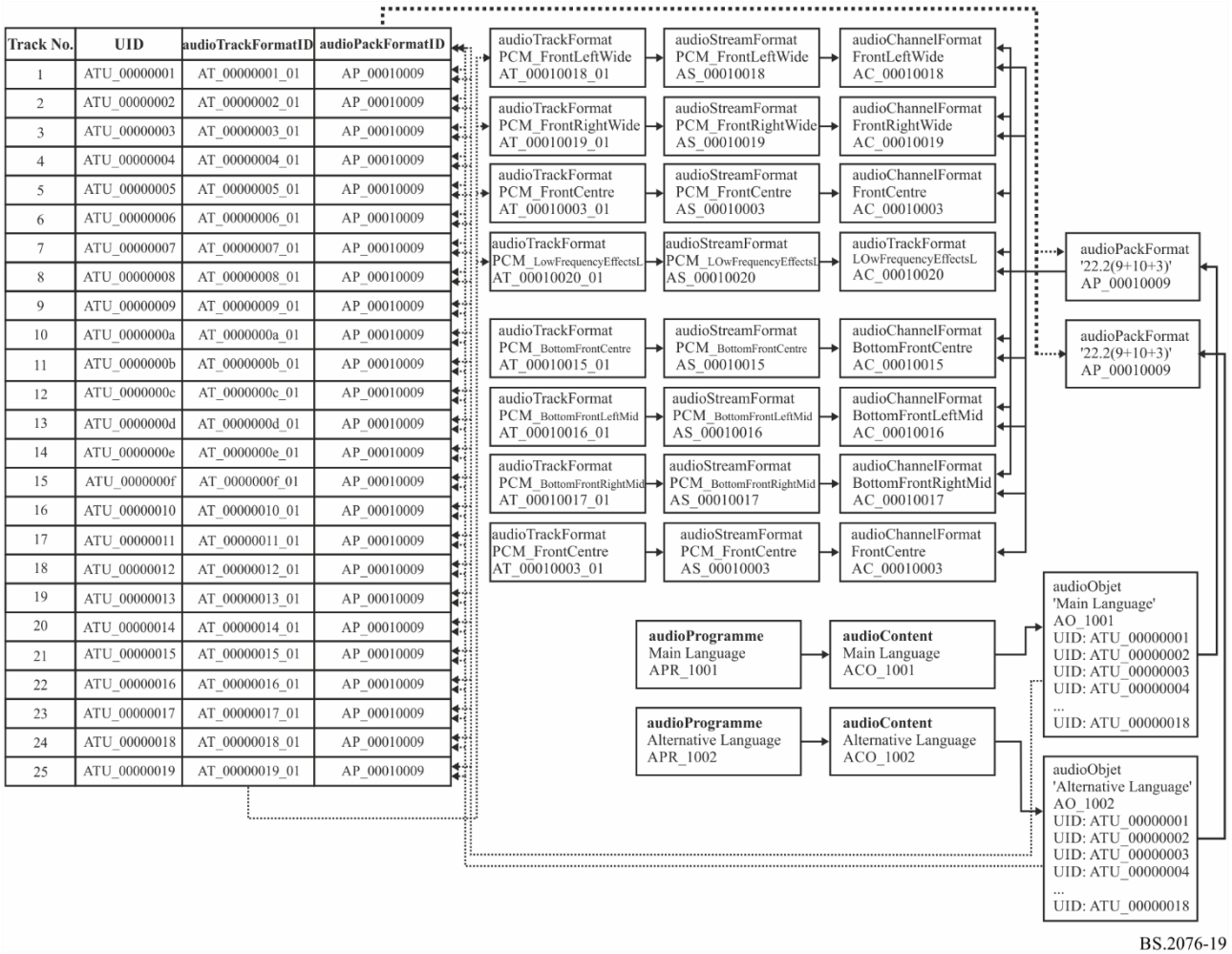
| | | | |
|---|---|---|---|
| audioObject | AO_1001 | MainLanguage | Object for 'MainLanguage', 22.2 format |
| audioObject | AO_1002 | AlternativeLanguage | Object for 'AlternativeLanguage', 22.2 format |
| audioContent | ACO_1001 | MainLanguage | 'MainLanguage' content |
| audioContent | ACO_1002 | AlternativeLanguage | 'AlternativeLanguage' content |
| audioProgramme | APR_1001 | MainLanguage | Programme 'MainLanguage' containing 'MainLanguage' content |
| audioProgramme | APR_1002 | AlternativeLanguage | Programme 'AlternativeLanguage' containing 'AlternativeLanguage' content |

## 6.2 Element relationships

The diagram in Fig. 19 shows how the defined elements relate to each other. The top half of the diagram covers the elements that describe the 22.2 channel and the one alternative dialogue object. The <*chna*> chunk in the middle shows how the tracks are connected to the format definitions. The content definition elements are at the bottom of the diagram, with the audioObject element containing the track UID references to the UID in the <*chna*> chunk.

FIGURE 19

**22-channel example diagram**




## 6.3 Sample code

This XML sample code does not include the audioFormatExtended parent element and the XML header for clarity. The excerpt of code covers both the format and content elements:

```
<!-- ############ -->
<!-- PROGRAMMES -->
<!-- ############ -->

<audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Main_Language">
  <audioContentIDRef>ACO_1001</audioContentIDRef>
</audioProgramme>

<audioProgramme audioProgrammeID="APR_1002" audioProgrammeName="Alternative_Language">
  <audioContentIDRef>ACO_1002</audioContentIDRef>
</audioProgramme>



<!-- ############ -->
<!-- CONTENTS -->
<!-- ############ -->
```

```
<audioContent audioContentID="ACO_1001" audioContentName="Main_Language">
  <audioObjectIDRef>AO_1001</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-24.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<audioContent audioContentID="ACO_1002" audioContentName="Alternative_Language">
  <audioObjectIDRef>AO_1002</audioObjectIDRef>
  <loudnessMetadata>
    <integratedLoudness>-24.0</integratedLoudness>
  </loudnessMetadata>
</audioContent>

<!-- ############ -->
<!-- OBJECTS -->
<!-- ############ -->

<audioObject audioObjectID="AO_1001" audioObjectName="Main_Language">
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000003</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000004</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000005</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000006</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000007</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000008</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000009</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000a</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000b</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000c</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000d</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000e</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000f</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000010</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000011</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000012</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000013</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000014</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000015</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000016</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000017</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000018</audioTrackUIDRef>
</audioObject>

<audioObject audioObjectID="AO_1002" audioObjectName="Alternative_Language">
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000019</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000004</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000005</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000006</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000007</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000008</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000009</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000a</audioTrackUIDRef>
```

```
  <audioTrackUIDRef>ATU_0000000b</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000c</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000d</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000e</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_0000000f</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000010</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000011</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000012</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000013</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000014</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000015</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000016</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000017</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000018</audioTrackUIDRef>
</audioObject>

<!-- ############ -->
<!-- PACKS -->
<!-- ############ -->

<audioPackFormat audioPackFormatID="AP_00010009" audioPackFormatName="22.2"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioChannelFormatIDRef>AC_00010018</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010019</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010003</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010020</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_0001001c</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_0001001d</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010009</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010021</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_0001000a</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_0001000b</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010022</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010023</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_0001000e</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_0001000c</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_0001001e</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_0001001f</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010013</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010014</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010011</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010015</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010016</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00010017</audioChannelFormatIDRef>
</audioPackFormat>



<!-- ############ -->
<!-- CHANNELS -->
<!-- ############ -->

<audioChannelFormat audioChannelFormatID="AC_00010018"
audioChannelFormatName="FrontLeftWide" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010018_00000001">
    <speakerLabel>M+060</speakerLabel>
    <position coordinate="azimuth">60.0</position>
```

```
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010019"
audioChannelFormatName="FrontRightWide" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010019_00000001">
    <speakerLabel>M-060</speakerLabel>
    <position coordinate="azimuth">-60.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010003"
audioChannelFormatName="FrontCentre" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010003_00000001">
    <speakerLabel>M+000</speakerLabel>
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010020" audioChannelFormatName="LFE1"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <frequency typeDefinition="lowPass">200</frequency>
  <audioBlockFormat audioBlockFormatID="AB_00010020_00000001">
    <speakerLabel>LFE1</speakerLabel>
    <position coordinate="azimuth">45.0</position>
    <position coordinate="elevation">-30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_0001001c"
audioChannelFormatName="BackLeftMid" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_0001001c_00000001">
    <speakerLabel>M+135</speakerLabel>
    <position coordinate="azimuth">135.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_0001001d"
audioChannelFormatName="BackRightMid" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_0001001d_00000001">
    <speakerLabel>M-135</speakerLabel>
    <position coordinate="azimuth">-135.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010001" audioChannelFormatName="FrontLeft"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010001_00000001">
```

```
    <speakerLabel>M+030</speakerLabel>
    <position coordinate="azimuth">30.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010002"
audioChannelFormatName="FrontRight" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010002_00000001">
    <speakerLabel>M-030</speakerLabel>
    <position coordinate="azimuth">-30.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010009"
audioChannelFormatName="BackCentre" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010009_00000001">
    <speakerLabel>M+180</speakerLabel>
    <position coordinate="azimuth">180.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010021" audioChannelFormatName="LFE2"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <frequency typeDefinition="lowPass">200</frequency>
  <audioBlockFormat audioBlockFormatID="AB_00010021_00000001">
    <speakerLabel>LFE2</speakerLabel>
    <position coordinate="azimuth">-45.0</position>
    <position coordinate="elevation">-30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_0001000a" audioChannelFormatName="SideLeft"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_0001000a_00000001">
    <speakerLabel>M+090</speakerLabel>
    <position coordinate="azimuth">90.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_0001000b" audioChannelFormatName="SideRight"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_0001000b_00000001">
    <speakerLabel>M-090</speakerLabel>
    <position coordinate="azimuth">-90.0</position>
    <position coordinate="elevation">0.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>
```

```
<audioChannelFormat audioChannelFormatID="AC_00010022"
audioChannelFormatName="TopFrontLeftMid" typeLabel="0001"
typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010022_00000001">
    <speakerLabel>U+045</speakerLabel>
    <position coordinate="azimuth">45.0</position>
    <position coordinate="elevation">30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010023"
audioChannelFormatName="TopFrontRightMid" typeLabel="0001"
typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010023_00000001">
    <speakerLabel>U-045</speakerLabel>
    <position coordinate="azimuth">-45.0</position>
    <position coordinate="elevation">30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_0001000e"
audioChannelFormatName="TopFrontCentre" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_0001000e_00000001">
    <speakerLabel>U+000</speakerLabel>
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">45.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_0001000c" audioChannelFormatName="TopCentre"
typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_0001000c_00000001">
    <speakerLabel>T+000</speakerLabel>
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">90.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_0001001e"
audioChannelFormatName="TopBackLeftMid" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_0001001e_00000001">
    <speakerLabel>U+135</speakerLabel>
    <position coordinate="azimuth">135.0</position>
    <position coordinate="elevation">30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_0001001f"
audioChannelFormatName="TopBackRightMid" typeLabel="0001"
typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_0001001f_00000001">
    <speakerLabel>U-135</speakerLabel>
    <position coordinate="azimuth">-135.0</position>
    <position coordinate="elevation">30.0</position>
    <position coordinate="distance">1.0</position>
```

```
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010013"
audioChannelFormatName="TopSideLeft" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010013_00000001">
    <speakerLabel>U+090</speakerLabel>
    <position coordinate="azimuth">90.0</position>
    <position coordinate="elevation">30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010014"
audioChannelFormatName="TopSideRight" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010014_00000001">
    <speakerLabel>U-090</speakerLabel>
    <position coordinate="azimuth">-90.0</position>
    <position coordinate="elevation">30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010011"
audioChannelFormatName="TopBackCentre" typeLabel="0001" typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010011_00000001">
    <speakerLabel>U+180</speakerLabel>
    <position coordinate="azimuth">180.0</position>
    <position coordinate="elevation">45.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010015"
audioChannelFormatName="BottomFrontCentre" typeLabel="0001"
typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010015_00000001">
    <speakerLabel>B+000</speakerLabel>
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">-30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010016"
audioChannelFormatName="BottomFrontLeftMid" typeLabel="0001"
typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010016_00000001">
    <speakerLabel>B+045</speakerLabel>
    <position coordinate="azimuth">45.0</position>
    <position coordinate="elevation">-30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00010017"
audioChannelFormatName="BottomFrontRightMid" typeLabel="0001"
typeDefinition="DirectSpeakers">
  <audioBlockFormat audioBlockFormatID="AB_00010017_00000001">
    <speakerLabel>B-045</speakerLabel>
```

```
    <position coordinate="azimuth">-45.0</position>
    <position coordinate="elevation">-30.0</position>
    <position coordinate="distance">1.0</position>
  </audioBlockFormat>
</audioChannelFormat>

<!-- ############ -->
<!-- STREAMS -->
<!-- ############ -->

<audioStreamFormat audioStreamFormatID="AS_00010018"
audioStreamFormatName="PCM_FrontLeftWide" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010018</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010018_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010019"
audioStreamFormatName="PCM_FrontRightWide" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010019</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010019_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010003"
audioStreamFormatName="PCM_FrontCentre" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010003</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010003_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010020" audioStreamFormatName="PCM_LFE1"
formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010020</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010020_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_0001001c"
audioStreamFormatName="PCM_BackLeftMid" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_0001001c</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_0001001c_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_0001001d"
audioStreamFormatName="PCM_BackRightMid" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_0001001d</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_0001001d_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010001"
audioStreamFormatName="PCM_FrontLeft" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010002"
audioStreamFormatName="PCM_FrontRight" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010002</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010009"
audioStreamFormatName="PCM_BackCentre" formatLabel="0001" formatDefinition="PCM">
```

```
  <audioChannelFormatIDRef>AC_00010009</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010009_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010021" audioStreamFormatName="PCM_LFE2"
formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010021</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010021_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_0001000a" audioStreamFormatName="PCM_SideLeft"
formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_0001000a</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_0001000a_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_0001000b"
audioStreamFormatName="PCM_SideRight" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_0001000b</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_0001000b_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010022"
audioStreamFormatName="PCM_TopFrontLeftMid" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010022</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010022_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010023"
audioStreamFormatName="PCM_TopFrontRightMid" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010023</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010023_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_0001000e"
audioStreamFormatName="PCM_TopFrontCentre" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_0001000e</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_0001000e_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_0001000c"
audioStreamFormatName="PCM_TopCentre" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_0001000c</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_0001000c_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_0001001e"
audioStreamFormatName="PCM_TopBackLeftMid" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_0001001e</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_0001001e_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_0001001f"
audioStreamFormatName="PCM_TopBackRightMid" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_0001001f</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_0001001f_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010013"
audioStreamFormatName="PCM_TopSideLeft" formatLabel="0001" formatDefinition="PCM">
```

```
  <audioChannelFormatIDRef>AC_00010013</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010013_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010014"
audioStreamFormatName="PCM_TopSideRight" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010014</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010014_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010011"
audioStreamFormatName="PCM_TopBackCentre" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010011</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010011_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010015"
audioStreamFormatName="PCM_BottomFrontCentre" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010015</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010015_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010016"
audioStreamFormatName="PCM_BottomFrontLeftMid" formatLabel="0001" formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010016</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010016_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00010017"
audioStreamFormatName="PCM_BottomFrontRightMid" formatLabel="0001"
formatDefinition="PCM">
  <audioChannelFormatIDRef>AC_00010017</audioChannelFormatIDRef>
  <audioTrackFormatIDRef>AT_00010017_01</audioTrackFormatIDRef>
</audioStreamFormat>

<!-- ############ -->
<!-- AUDIO TRACKS -->
<!-- ############ -->

<audioTrackFormat audioTrackFormatID="AT_00010018_01"
audioTrackFormatName="PCM_FrontLeftWide" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010018</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010019_01"
audioTrackFormatName="PCM_FrontRightWide" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010019</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010003_01"
audioTrackFormatName="PCM_FrontCentre" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010003</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010020_01" audioTrackFormatName="PCM_LFE1"
formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010020</audioStreamFormatIDRef>
</audioTrackFormat>
```

```
<audioTrackFormat audioTrackFormatID="AT_0001001c_01"
audioTrackFormatName="PCM_BackLeftMid" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_0001001c</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_0001001d_01"
audioTrackFormatName="PCM_BackRightMid" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_0001001d</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010001_01"
audioTrackFormatName="PCM_FrontLeft" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010001</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010002_01"
audioTrackFormatName="PCM_FrontRight" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010002</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010009_01"
audioTrackFormatName="PCM_BackCentre" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010009</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010021_01" audioTrackFormatName="PCM_LFE2"
formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010021</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_0001000a_01" audioTrackFormatName="PCM_SideLeft"
formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_0001000a</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_0001000b_01"
audioTrackFormatName="PCM_SideRight" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_0001000b</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010022_01"
audioTrackFormatName="PCM_TopFrontLeftMid" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010022</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010023_01"
audioTrackFormatName="PCM_TopFrontRightMid" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010023</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_0001000e_01"
audioTrackFormatName="PCM_TopFrontCentre" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_0001000e</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_0001000c_01"
audioTrackFormatName="PCM_TopCentre" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_0001000c</audioStreamFormatIDRef>
</audioTrackFormat>
```

```
<audioTrackFormat audioTrackFormatID="AT_0001001e_01"
audioTrackFormatName="PCM_TopBackLeftMid" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_0001001e</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_0001001f_01"
audioTrackFormatName="PCM_TopBackRightMid" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_0001001f</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010013_01"
audioTrackFormatName="PCM_TopSideLeft" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010013</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010014_01"
audioTrackFormatName="PCM_TopSideRight" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010014</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010011_01"
audioTrackFormatName="PCM_TopBackCentre" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010011</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010015_01"
audioTrackFormatName="PCM_BottomFrontCentre" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010015</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010016_01"
audioTrackFormatName="PCM_BottomFrontLeftMid" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010016</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00010017_01"
audioTrackFormatName="PCM_BottomFrontRightMid" formatLabel="0001" formatDefinition="PCM">
  <audioStreamFormatIDRef>AS_00010017</audioStreamFormatIDRef>
</audioTrackFormat>


<!-- ############ -->
<!-- AUDIO TRACK UIDs -->
<!-- ############ -->

<audioTrackUID UID="ATU_00000001">
  <audioTrackFormatIDRef>AT_00010018_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000002">
  <audioTrackFormatIDRef>AT_00010019_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000003">
  <audioTrackFormatIDRef>AT_00010003_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>
```

```
<audioTrackUID UID="ATU_00000004">
  <audioTrackFormatIDRef>AT_00010020_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000005">
  <audioTrackFormatIDRef>AT_0001001c_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000006">
  <audioTrackFormatIDRef>AT_0001001d_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000007">
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000008">
  <audioTrackFormatIDRef>AT_00010002_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000009">
  <audioTrackFormatIDRef>AT_00010009_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_0000000a">
  <audioTrackFormatIDRef>AT_00010021_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_0000000b">
  <audioTrackFormatIDRef>AT_0001000a_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_0000000c">
  <audioTrackFormatIDRef>AT_0001000b_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_0000000d">
  <audioTrackFormatIDRef>AT_00010022_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_0000000e">
  <audioTrackFormatIDRef>AT_00010023_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_0000000f">
  <audioTrackFormatIDRef>AT_0001000e_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
```

```
</audioTrackUID>

<audioTrackUID UID="ATU_00000010">
  <audioTrackFormatIDRef>AT_0001000c_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000011">
  <audioTrackFormatIDRef>AT_0001001e_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000012">
  <audioTrackFormatIDRef>AT_0001001f_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000013">
  <audioTrackFormatIDRef>AT_00010013_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000014">
  <audioTrackFormatIDRef>AT_00010014_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000015">
  <audioTrackFormatIDRef>AT_00010011_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000016">
  <audioTrackFormatIDRef>AT_00010015_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000017">
  <audioTrackFormatIDRef>AT_00010016_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>

<audioTrackUID UID="ATU_00000018">
  <audioTrackFormatIDRef>AT_00010017_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>
```

```
<audioTrackUID UID="ATU_00000019">
  <audioTrackFormatIDRef>AT_00010003_01</audioTrackFormatIDRef>
  <audioPackFormatIDRef>AP_00010009</audioPackFormatIDRef>
</audioTrackUID>
```

## 7 Example of the use of the Matrix type

The example illustrates both an encoding and decoding matrices that are associated with each other, in this case the 5.1 to Lo/Ro downmix matrix. The audio tracks are the Lo/Ro channels, so the decoding matrix describes how these are converted back to channel-based channels (trivially in this case), and the encoding matrix that was used to produce these tracks.

In reality, an Lo/Ro downmix would more likely be specified using a single direct matrix, as the Lo/Ro channels are effectively channel-based. This example is used to illustrate the concept of an encoding and decoding matrix pair, where the decoding matrix is just a trivial identity matrix.

### 7.1 Summary of elements

These are the elements in the format part of the description:

TABLE 75

**Matrix example format elements**

| Element | ID | Name | Description |
|---|---|---|---|
| audioTrackFormat | AT_00021103_01 | PCM_Lo/Ro_Decode_Left | Defines track as PCM |
| audioTrackFormat | AT_00021104_01 | PCM_Lo/Ro_Decode_Right | Defines track as PCM |
| audioStreamFormat | AS_00021103 | PCM_Lo/Ro_Decode_Left | Defines stream as PCM |
| audioStreamFormat | AS_00021104 | PCM_Lo/Ro_Decode_Right | Defines stream as PCM |
| audioChannelFormat and audioBlockFormat | AC_00021003 AB_00021003_00000001 | Lo/Ro_Left | Describes channel as Lo matrix encoding |
| audioChannelFormat and audioBlockFormat | AC_00021004 AB_00021004_00000001 | Lo/Ro_Right | Describes channel as Ro matrix encoding |
| audioChannelFormat and audioBlockFormat | AC_00021103 AB_00021103_00000001 | Lo/Ro_Decode_Left | Describes channel as Lo matrix decoding |
| audioChannelFormat and audioBlockFormat | AC_00021104 AB_00021104_00000001 | Lo/Ro_Decode_Right | Describes channel as Ro matrix decoding |
| audioPackFormat | AP_00021002 | Lo/Ro | Defines a Lo/Ro pack encoding matrix (from 5.1 channels). |
| audioPackFormat | AP_00021102 | Lo/Ro_Decode | Defines a Lo/Ro pack decoding matrix (to 2 channels). |

These are the elements in the content part of the description:

TABLE 76

**Matrix example content elements**

| Element | ID | Name | Description |
|---|---|---|---|
| audioObject | AO_1001 | Lo/Ro_Downmix | Object for Lo/Ro encoded channels |

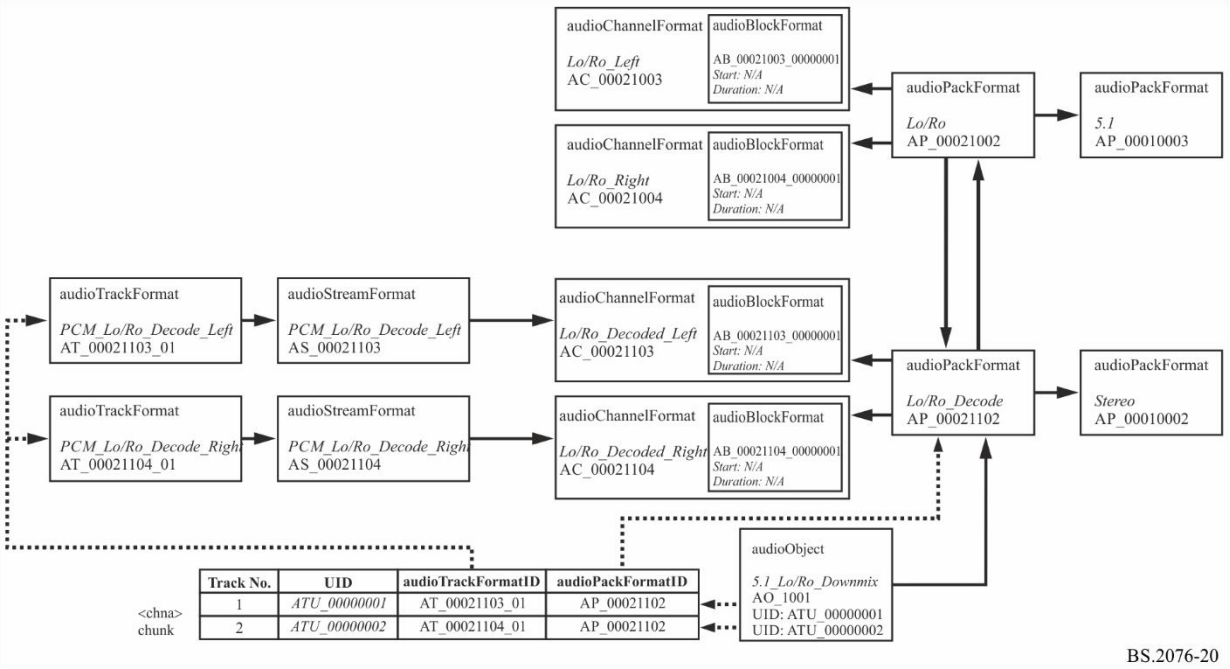
## 7.2 Element Relationships

The diagram in Fig. 20 shows how the defined elements relate to each other. The two audioTrackFormat and audioStreamFormat elements refer to audioChannelFormats that describe a decoding matrix. These are refered from an audioPackFormat element that describes the while decoding matrix. This audioPackFormat element also references another audioPackFormat element that describes an associated encoding matrix (which in turn references two encoding matrix audioChannelFormat elements). Each of the matrix audioPackFormat elements also reference 'DirectSpeakers' audioPackFormat elements, which are not included in the XML as they are common definitions (hence greyed out in the diagram).

The <*chna*> chunk at the bottom shows how the tracks are connected to the format definitions. The audioObject element containing the track UID references to the UID in the <*chna*> chunk, and references the decoding matrix audioPackFormat element.

FIGURE 20

**Matrix example diagram**




## 7.3 Sample code

This XML sample code does not include the audioFormatExtended parent element and the XML header for clarity. The elements that are in the common definitions (ITU-R BS.2094) have also been removed for clarity. The code contains both the content and format parts, but omits the common definition elements that are referenced:

```
<!-- ############ -->
<!-- OBJECTS -->
<!-- ############ -->

<audioObject audioObjectID="AO_1001" audioObjectName="Lo/Ro_Downmix">
    <audioPackFormatIDRef>AP_00021102</audioPackFormatIDRef>
    <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
    <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
</audioObject>
```

```
<!-- ############ -->
<!-- PACKS -->
<!-- ############ -->

<audioPackFormat audioPackFormatID="AP_00021002" audioPackFormatName="Lo/Ro"
typeLabel="0002" typeDefinition="Matrix">
  <decodePackFormatIDRef>AP_00021102</decodePackFormatIDRef>
  <inputPackFormatIDRef>AP_00010003</inputPackFormatIDRef>
  <audioChannelFormatIDRef>AC_00021003</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00021004</audioChannelFormatIDRef>
</audioPackFormat>

<audioPackFormat audioPackFormatID="AP_00021102" audioPackFormatName="Lo/Ro_Decode"
typeLabel="0002" typeDefinition="Matrix">
  <encodePackFormatIDRef>AP_00021002</encodePackFormatIDRef>
  <outputPackFormatIDRef>AP_00010002</outputPackFormatIDRef>
  <audioChannelFormatIDRef>AC_00021103</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00021104</audioChannelFormatIDRef>
</audioPackFormat>

<!-- ############ -->
<!-- CHANNELS -->
<!-- ############ -->

<audioChannelFormat audioChannelFormatID="AC_00021003"
audioChannelFormatName="Lo/Ro_Left" typeLabel="0002" typeDefinition="Matrix">
  <audioBlockFormat audioBlockFormatID="AB_00021003_00000001">
    <matrix>
      <coefficient gain="1.0">AC_00010001</coefficient>
      <coefficient gain="cvar">AC_00010003</coefficient>
      <coefficient gain="svar">AC_00010005</coefficient>
    </matrix>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00021004"
audioChannelFormatName="Lo/Ro_Right" typeLabel="0002" typeDefinition="Matrix">
  <audioBlockFormat audioBlockFormatID="AB_00021004_00000001">
    <matrix>
      <coefficient gain="1.0">AC_00010002</coefficient>
      <coefficient gain="cvar">AC_00010003</coefficient>
      <coefficient gain="svar">AC_00010006</coefficient>
    </matrix>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00021103"
audioChannelFormatName="Lo/Ro_Decode_Left" typeLabel="0002" typeDefinition="Matrix">
  <audioBlockFormat audioBlockFormatID="AB_00021103_00000001">
    <outputChannelFormatIDRef>AC_00010001</outputChannelFormatIDRef>
    <matrix>
      <coefficient gain="1.0">AC_00021003</coefficient>
    </matrix>
  </audioBlockFormat>
</audioChannelFormat>

<audioChannelFormat audioChannelFormatID="AC_00021104"
audioChannelFormatName="Lo/Ro_Decode_Right" typeLabel="0002" typeDefinition="Matrix">
```

```
  <audioBlockFormat audioBlockFormatID="AB_00021104_00000001">
    <outputChannelFormatIDRef>AC_00010002</outputChannelFormatIDRef>
    <matrix>
      <coefficient gain="1.0">AC_00021004</coefficient>
    </matrix>
  </audioBlockFormat>
</audioChannelFormat>

<!-- ############ -->
<!-- STREAMS -->
<!-- ############ -->

<audioStreamFormat audioStreamFormatID="AS_00021103"
audioStreamFormatName="PCM_Lo/Ro_Deocde_Left" formatLabel="0001"
formatDefinition="PCM">
    <audioChannelFormatIDRef>AC_00021103</audioChannelFormatIDRef>
    <audioTrackFormatIDRef>AT_00021103_01</audioTrackFormatIDRef>
</audioStreamFormat>

<audioStreamFormat audioStreamFormatID="AS_00021104"
audioStreamFormatName="PCM_Lo/Ro_Decode_Right" formatLabel="0001"
formatDefinition="PCM">
    <audioChannelFormatIDRef>AC_00021104</audioChannelFormatIDRef>
    <audioTrackFormatIDRef>AT_00021104_01</audioTrackFormatIDRef>
</audioStreamFormat>

<!-- ############ -->
<!-- AUDIO TRACKS -->
<!-- ############ -->

<audioTrackFormat audioTrackFormatID="AT_00021103_01"
audioTrackFormatName="PCM_Lo/Ro_Decode_Left" formatLabel="0001" formatDefinition="PCM">
    <audioStreamFormatIDRef>AS_00021103</audioStreamFormatIDRef>
</audioTrackFormat>

<audioTrackFormat audioTrackFormatID="AT_00021104_01"
audioTrackFormatName="PCM_Lo/Ro_Decode_Right" formatLabel="0001"
formatDefinition="PCM">
    <audioStreamFormatIDRef>AS_00021104</audioStreamFormatIDRef>
</audioTrackFormat>
```