INTERNATIONAL STANDARD

ISO/IEC 15938-4

First edition
2002-06-15

Information technology — Multimedia content description interface —

Part 4: **Audio**

Technologies de l'information — Interface de description du contenu multimédia —

Partie 4: Audio

Reference number
ISO/IEC 15938-4:2002(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

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

Printed in Switzerland

Contents

Page

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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

ISO/IEC 15938-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 15938 consists of the following parts, under the general title *Information technology — Multimedia content description interface*:

— *Part 1: Systems*

— *Part 2: Description definition language*

— *Part 3: Visual*

— *Part 4: Audio*

— *Part 5: Multimedia description schemes*

— *Part 6: Reference software*

— *Part 7: Conformance testing*

— *Part 8: Extraction and use of MPEG-7 descriptions*

Annexes A and B of this part of ISO/IEC 15938 are for information only.

Introduction

This standard, also known as "Multimedia Content Description Interface," provides a standardized set of technologies for describing multimedia content. The standard addresses a broad spectrum of multimedia applications and requirements by providing a metadata system for describing the features of multimedia content.

The following are specified in this standard:

- **Description Schemes (DS)** describe entities or relationships pertaining to multimedia content. Description Schemes specify the structure and semantics of their components, which may be Description Schemes, Descriptors, or datatypes.

- **Descriptors (D)** describe features, attributes, or groups of attributes of multimedia content.

- **Datatypes** are the basic reusable datatypes employed by Description Schemes and Descriptors

- **Description Definition Language (DDL)** defines Description Schemes, Descriptors, and Datatypes by specifying their syntax, and allows their extension.

- **Systems tools** support delivery of descriptions, multiplexing of descriptions with multimedia content, synchronization, file format, and so forth.

This standard is subdivided into eight parts:

Part 1 – Systems: specifies the tools for preparing descriptions for efficient transport and storage, compressing descriptions, and allowing synchronization between content and descriptions.

Part 2 – Description definition language: specifies the language for defining the standard set of description tools (DSs, Ds, and datatypes) and for defining new description tools.

Part 3 – Visual: specifies the description tools pertaining to visual content.

Part 4 – Audio: specifies the description tools pertaining to audio content.

Part 5 – Multimedia description schemes: specifies the generic description tools pertaining to multimedia including audio and visual content.

Part 6 – Reference software: provides a software implementation of the standard.

Part 7 – Conformance testing: specifies the guidelines and procedures for testing conformance of implementations of the standard.

Part 8 – Extraction and use of MPEG-7 descriptions: provides guidelines and examples of the extraction and use of descriptions.

Information technology — Multimedia content description interface —

Part 4: Audio

1 Scope

1.1 Definition of Scope

This International Standard defines a Multimedia Content Description Interface, specifying a series of interfaces from system to application level to allow disparate systems to interchange information about multimedia content. It describes the architecture for systems, a language for extensions and specific applications, description tools in the audio and visual domains, as well as tools that are not specific to audio-visual domains. As a whole, this International Standard encompassing all of the aforementioned components is known as “MPEG-7.” MPEG-7 is divided into eight parts (as defined in the Foreword).

This part of the MPEG-7 Standard (Part 4: Audio) specifies description tools that pertain to multimedia in the audio domain. See below for further details of application.

This part of the MPEG-7 Standard is intended to be implemented in conjunction with other parts of the standard. In particular, MPEG-7 Part 4: Audio assumes knowledge of Part 2: Description Definition Language (DDL) in its normative syntactic definitions of Descriptors and Description Schemes. This part of the standard also has dependencies upon clauses in Part 5: Multimedia Description Schemes, namely many of the fundamental Description Schemes that extend the basic type capabilities of the DDL.

MPEG-7 is an extensible standard. The method to extend the standard beyond the Description Schemes provided in the standard is to define new ones in the DDL, and to make those DSs available with the instantiated descriptions. Further details are available in Part 2. To avoid duplicate functionality with other parts of the standard, the DDL is the only extension facility provided.

1.2 Fields of application

MPEG-7 Part 4: Audio is applicable to all forms of audio content. The encoding format or medium of the said audio is not limited in any way, and may include audio held in an analogue medium such as magnetic tape or optical film. The content of the audio is not limited within or without music, speech, sound effects, soundtracks, or any mixtures thereof.

The tools listed in this part of the International Standard are applicable to both audio in isolation and to audio associated with video.

The specific tools provided within the Audio portion of the standard are designed to work in conjunction with the Multimedia Description Schemes that apply to both audio and video. Because of the “toolbox” nature of the standard, the most appropriate tools from the different parts of the standard may be mixed, within the constraints of the DDL.

The MPEG-7 Audio tools are applicable to two general areas: low-level audio description, in the case of the Audio Framework (clause 5), and application-driven description, in the case of the High Level Tools (clause 6).

The Audio Framework tools are applicable to general audio, without regard to the specific content carried by the encoded signal. The Scalable Series provides general capabilities for multi-level sampled data. The Audio Description Framework defines specific descriptors for use with the Scalable Series or with Audio Segments, which has properties inherited from the general Segment described in the Multimedia Description Schemes part of the standard. The Silence Descriptor works with the Segment descriptor, and is applicable across all possible audio signals.

The high level description tools are applicable to specific types of content within audio. The specific domains are well documented within the introduction to each sub-clause. The audio domains encompassed by the various MPEG-7 Audio tools are speech, sound effects, musical instruments, melodies within music and general audio recognition. These specialised tools may be employed in conjunction with the other tools within the standard.

2 Terms and definitions

For the purposes of this part of ISO/IEC 15938, the following terms and definitions apply.

2.1
Frame

A Frame is defined as a short period of time of the signal on which the instantaneous analysis if perform. For a signal, noted $s(t)$ (in continuous time noted t), and for an analysis window of type hamming, noted $h(t)$ and of temporal length L, the f^{th} signal frame is defined as

$$x(f,t) = s(t) \times h(t - f \times S)$$

where S is the hop size

2.2
Hop size

The hop size defines the temporal distance between two successive analyses

2.3
Running window analysis

A running window analysis is an analysis obtained by multiplying the signal by a window function which is shifted along time by integer multiple of a parameter called the hop size. For a window function $h(t)$, and a hop size S, the f^{th} shifting of the window is equal to $h(t - fS)$.

2.4
Instantaneous values

The instantaneous value of a (Timbre) descriptor based peak estimation is defined to be the result of analysis on a frame level. The global value of a (Timbre) descriptor based on peak estimation is defined to be the average over all frames of the segment of the instantaneous value.

3 Symbols and abbreviated terms

—	ASR	Automatic Speech Recognition
—	CPU	Central Processing Unit
—	D	Descriptor
—	DC	Direct Current (0 Hz)
—	DDL	Description Definition Language
—	DFT	Discrete Fourier Transform
—	DS	Description Scheme

—	FFT	Fast Fourier Transform
—	HMM	Hidden Markov Model
—	Hz	Hertz, frequency in cycles per second
—	LLD	Low Level Descriptor
—	log	Logarithm (unspecified base)
—	LPC	Linear Predictive Coding
—	MSD	Maximum Squared Distance (from the mean)
—	OOV	Out of Vocabulary, describing a word that is not in the vocabulary of an automatic speech recogniser
—	RMS	Root Mean Square
—	SR	Sample Rate
—	STFT	Short Time Fourier Transform
—	XML	Extensible Markup Language

4 Conventions

4.1 Description Definition Language

All DDL in this document is defined in a single namespace. The schema wrapper is assumed to begin

```
<schema targetNamespace="urn:mpeg:mpeg7:schema:2001"
        xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xml="http://www.w3.org/XML/1998/namespace"
        elementFormDefault="qualified" attributeFormDefault="unqualified">
```

and end

```
</schema>
```

Under this definition, the default namespace in a schema definition document is specified as XML Schema and thus a prefix `xsd:` is not needed. Instead, references to the element and types defined in the MPEG-7 schema must be qualified with `mpeg7:` prefix. For example,

```
  <complexType name="MyElementType">
    <sequence>
      <element name="MyVector" type="mpeg7:MyVectorType"/>
    </sequence>
    <attribute name="myAttribute" type="mpeg7:unsigned8"/>
  </complexType>
```

4.2 Audio representation

Within the scope of this standard, the samples of the described audio signals are interpreted as two's complement fractional numbers (i.e. numbers between –1, inclusive, and +1, exclusive), where the Most Significant Bit (MSB) represents the value -1.

5 Audio Framework

5.1 Introduction

The Audio Framework contains low level tools designed to provide a basis for construction of higher level audio applications.

There are essentially two ways of describing low-level audio features. One may sample values at regular intervals or one may use `AudioSegment`s to demark regions of similarity and dissimilarity within the sound. Both of these possibilities are embodied in the low-level descriptor types, `AudioLLDScalarType` and `AudioLLDVectorType`. A descriptor of either of these types may be instantiated as sampled values in a `ScalableSeries`, or as a summary descriptor within an `AudioSegment`. `AudioSegment`, which is a concept that permeates the MPEG-7 Audio standard, is specified in ISO/IEC 15938 Part 5, Multimedia Description Schemes, but we also give a brief overview here.

An `AudioSegment` is a temporal interval of audio material, which may range from arbitrarily short intervals to the entire audio portion of a media document. A required element of an `AudioSegment` is a `MediaTime` descriptor that denotes the beginning and end of the segment. The `TemporalMask` DS is a construct that allows one to specify a temporally non-contiguous `AudioSegment`. An `AudioSegment` (as with any `SegmentType`) may be decomposed hierarchically to describe a tree of `Segment`s.

Another key concept is in the abstract datatypes: `AudioDType` and `AudioDSType`. In order for an audio descriptor or description scheme to be attached to a segment, it must inherit from one of these two types. They are defined in ISO/IEC 15938 part 5. The relationship between these types is shown in Figure 1.

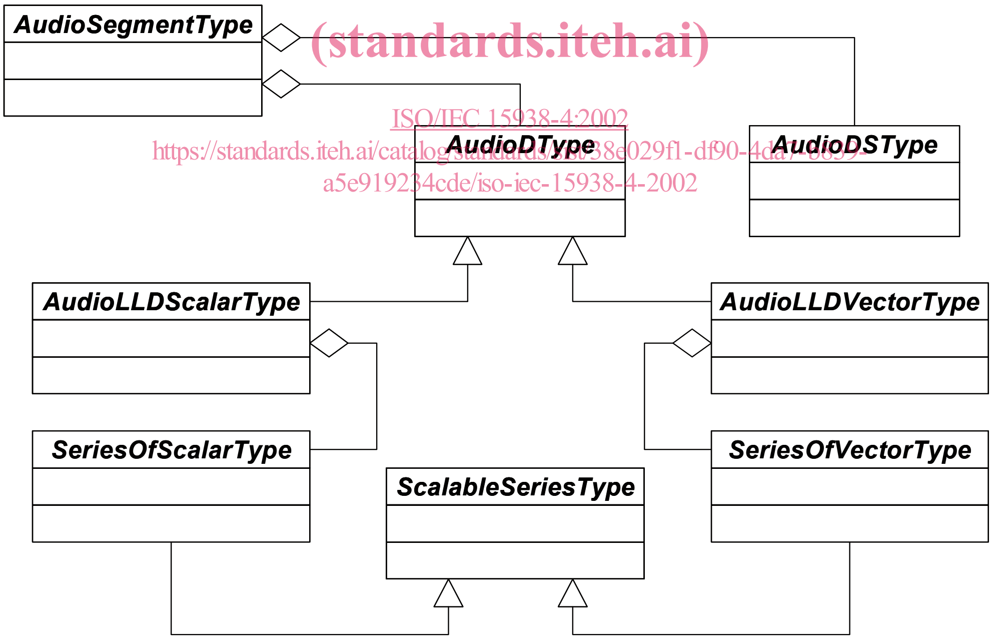

Figure 1 — Illustration of the various structural types in the Audio Framework

5.2 Scalable Series

5.2.1 Introduction

Scalable series are datatypes for series of values (scalars or vectors). They allow the series to be scaled (downsampled) in a well-defined fashion. Two types are available: `SeriesOfScalarType` and `SeriesOfVectorType`. They are useful in particular to build descriptors that contain *time series* of values.

5.2.2 ScalableSeriesType

This is an abstract type inherited by `SeriesOfScalarType` and `SeriesOfVectorType`. Its attributes define the dimensions and scaling ratio of the series.

5.2.2.1 Syntax

```
<!-- ################################################################### -->
<!-- Definition of ScalableSeries datatype                               -->
<!-- ################################################################### -->
<complexType name="ScalableSeriesType" abstract="true">
  <sequence>
    <element name="Scaling" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <attribute name="ratio" type="positiveInteger" use="required"/>
        <attribute name="numOfElements" type="positiveInteger"
                   use="required"/>
      </complexType>
    </element>
  </sequence>
  <attribute name="totalNumOfSamples" type="positiveInteger" use="required"/>
</complexType>
```

5.2.2.2 Semantics

Name	Definition
`ScalableSeriesType`	An abstract type representing series of values, at full resolution or after scaling (downsampling) by a scaling operation. In the latter case the series contains sequences that have been concatenated together. Within each sequence, the elements share the same scale ratio.
`Scaling`	To specify how the original samples are scaled. If absent, the original samples are described without scaling.
`ratio`	Scale ratio (number of original samples represented by each scaled sample) common to all elements in a sequence. The value to be used when `Scaling` is absent is 1.
`numOfElements`	Number of scaled elements in a sequence. The value to be used when `Scaling` is absent is equal to the value of `totalNumOfSamples`.
`totalNumOfSamples`	Total number of samples of the original series (before scaling).

Note that the last sample of the series may summarize fewer than `ratio` samples. This happens if `totalNumOfSamples` is smaller than the sum over runs of the product of `numOfElements` by `ratio`. An illustration of the Scalable Series is shown in Figure 2, where ‘k’ is an index in the scaled series. In this figure, the 31 samples of the original series (filled circles) are summarized by 13 samples of the scaled series (open circles). The first three scaled samples each summarizes two original samples, the next two six, the next two one, etc. The last scaled sample has nominally a ratio of two, but actually summarizes only one original sample. This situation is legal, and detected by comparing the sum of `ratio` times `numOfElements` products to `totalNumOfSamples`.

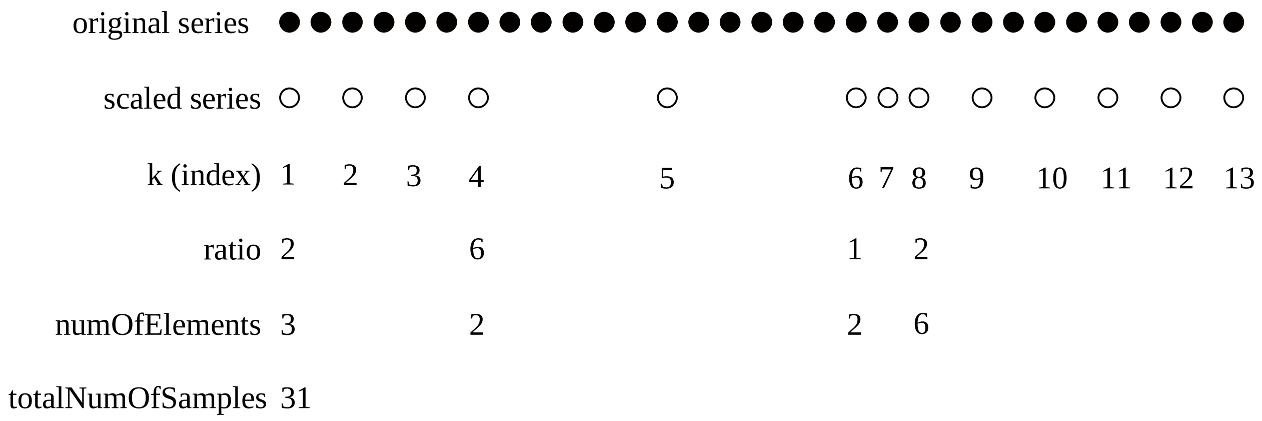

Figure 2 — An illustration of the scalable series

5.2.3 SeriesOfScalarType

This descriptor represents a series of scalars, at full resolution or scaled. Use this type within descriptor definitions to represent a series of feature values.

5.2.3.1 Syntax

```
<!-- ##################################################################### -->
<!-- Definition of SeriesOfScalar datatype                                 -->
<!-- ##################################################################### -->
<complexType name="SeriesOfScalarType">
  <complexContent>
    <extension base="mpeg7:ScalableSeriesType">
      <sequence>
        <element name="Raw" type="mpeg7:floatVector" minOccurs="0"/>
        <element name="Min" type="mpeg7:floatVector" minOccurs="0"/>
        <element name="Max" type="mpeg7:floatVector" minOccurs="0"/>
        <element name="Mean" type="mpeg7:floatVector" minOccurs="0"/>
        <element name="Random" type="mpeg7:floatVector" minOccurs="0"/>
        <element name="First" type="mpeg7:floatVector" minOccurs="0"/>
        <element name="Last" type="mpeg7:floatVector" minOccurs="0"/>
        <element name="Variance" type="mpeg7:floatVector" minOccurs="0"/>
        <element name="Weight" type="mpeg7:floatVector" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

5.2.3.2 Semantics

Name	Definition
`SeriesOfScalarType`	A representation of a series of scalar values of a feature.
`Raw`	Series of unscaled samples (full resolution). Use only if scaling is absent to indicate the entire series.
`Min`	Series of minima of groups of samples. The value of `numOfElements` shall equal the length of the vector. This element shall be absent or empty if the `Raw` element is present.

Name	Definition
Max	Series of maxima of groups of samples. The value of numOfElements shall equal the length of the vector. This element shall be absent or empty if the Raw element is present.
Mean	Series of means of groups of samples. The value of numOfElements shall equal the length of the vector. This element shall be absent or empty if the Raw element is present.
Random	Downsampled series (one sample selected at random from each group of samples). The value of numOfElements shall equal the length of the vector. This element shall be absent or empty if the Raw element is present.
First	Downsampled series (first sample selected from each group of samples). The value of numOfElements shall equal the length of the vector. This element shall be absent or empty if the Raw element is present.
Last	Downsampled series (last sample selected from each group of samples). The value of numOfElements shall equal the length of the vector. This element shall be absent or empty if the Raw element is present.
Variance	Series of variances of groups of samples. The value of numOfElements shall equal the length of the vector. This element shall be absent or empty if the Raw element is present. Mean must be present in order for Variance to be present.
Weight	Optional series of weights. Contrary to other fields, these do not represent values of the descriptor itself, but rather auxiliary weights to control scaling (see below). The value of numOfElements shall equal the length of the vector.

Note: Data of a full resolution series (ratio = 1) are stored in the Raw field. Accompanying zero-sized fields (such as Mean) indicate how the series may be scaled, if the need for scaling arises. The data are then stored in the scaled field(s) and the Raw field disappears.

Scalable Series allow data to be stored at reduced resolution, according to a number of possible scaling operations. The allowable operations are those that are *scalable* in the following sense. Suppose the original series is scaled by a scale ratio of P, and this scaled series is then rescaled by a factor of Q. The result is the same as if the original series had been scaled by a scale ratio of $N = PQ$.

Figure 3 illustrates the scalability property. This scaled series can be derived indifferently from the original series by applying the scaling operation with the ratios shown, or from the scaled Series of Figure 2 by applying the appropriate rescaling operation. The result is identical. Scaling operations are chosen among those for which this property can be enforced.

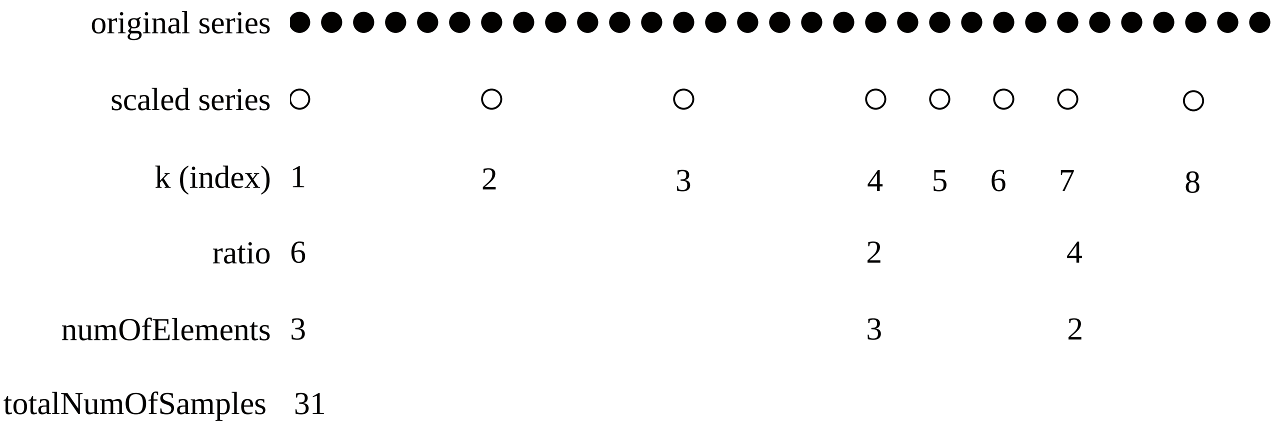

Figure 3 — An illustration of the scalability property

If the scaling operations are used, they shall be computed as follows.

Name	Definition	Definition if Weight present
Min	$m_k = \min_{i=1+(k-1)N}^{kN} x_i$	Ignore samples with zero weight. If all have zero weight, set to zero by convention.
Max	$M_k = \max_{i=1+(k-1)N}^{kN} x_i$	Ignore samples with zero weight. If all have zero weight, set to zero by convention.
Mean	$\bar{x}_k = (1/N) \sum_{i=1+(k-1)N}^{kN} x_i$	$\bar{x}_k = \sum_{i=1+(k-1)N}^{kN} w_i x_i \Big/ \sum_{i=1+(k-1)N}^{kN} w_i$ If all samples have zero weight, set to zero by convention.
Random	choose at random among N samples	Choose at random with probabilities proportional to weights. If all samples have zero weight, set to zero by convention.
First	choose the first of N samples	Choose first non-zero-weight sample. If all samples have zero weight, set to zero by convention.
Last	choose the last of N samples	Choose last non-zero-weight sample. If all samples have zero weight, set to zero by convention.
Variance	$z_k = (1/N) \sum_{i=1+(k-1)N}^{kN} (x_i - \bar{x}_k)^2$ $= (1/N) \sum_{i=1+(k-1)N}^{kN} x_i^2 - \bar{x}_k^2$	$z_k = \sum_{i=1+(k-1)N}^{kN} w_i (x_i - \bar{x}_k)^2 \Big/ \sum_{i=1+(k-1)N}^{kN} w_i$ If all samples have zero weight, set to zero by convention.
Weight	$\bar{w}_k = (1/N) \sum_{i=1+(k-1)N}^{kN} w_i$	

In these formulae, k is an index in the scaled series, and i an index in the original series. N is the number of samples summarized by each scaled sample. The formula for `Variance` differs from the standard formula for unbiased variance by the presence of N rather than $N-1$. Unbiased variance is easy to derive from it. If the `Weight` field is present, the terms of all sums are weighted.

5.2.4 SeriesOfScalarBinaryType

Use this type to instantiate a series of scalars with a uniform power-of-two ratio. The restriction to a power-of-two ratio eases the comparison of series with different `ratio`s as the decimation required for the comparison will also be a power of 2. Such decimation allows perfect comparison. It also allows an additional scaling operation to be defined (scalewise variance). Considering these computational properties of power-of-two scale ratios, the `SeriesOfScalarBinaryType` is the most useful of the Scalable Series family.

Note that the types `SeriesOfScalarBinaryType` and `SeriesOfVectorBinaryType` inherit from the appropriate non-binary type. This means that although they are not used directly in this document, they can be used in place of the non-binary type at any time.

Figure 4 shows an illustration of `SeriesOfScalarBinaryType`. In this subtype of `SeriesOfScalarType`, all `ratio`s must all be powers of two, and all `numOfElements` values must be such that the series may be rescaled to the largest `ratio` that it contains. In this example, the largest `ratio` is 8. The first four samples (`ratio` 2) can be rescaled to a single sample with `ratio` 8, and the last four samples can be rescaled to two samples with `ratio` 8. Note, as in previous illustrations, that the last scaled sample summarizes fewer original samples than its nominal `ratio`.

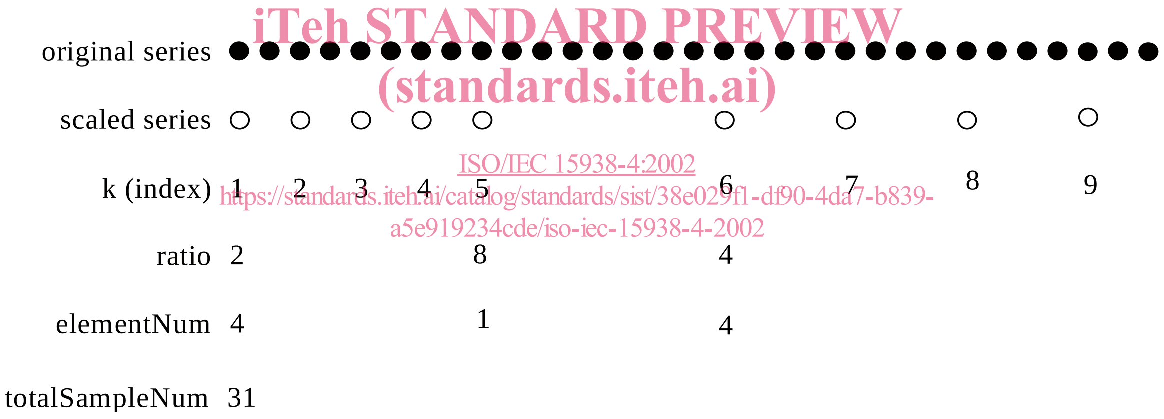

Figure 4 — An illustration of SeriesOfScalarBinaryType

5.2.4.1 Syntax

```
<!-- ##################################################################### -->
<!-- Definition of SeriesOfScalarBinary datatype                           -->
<!-- ##################################################################### -->
<complexType name="SeriesOfScalarBinaryType">
  <complexContent>
    <extension base="mpeg7:SeriesOfScalarType">
      <sequence>
        <element name="VarianceScalewise" type="mpeg7:FloatMatrixType"
                 minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```