
**Information technology — Multimedia
content description interface —**

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

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

Partie 8: Extraction et utilisation des descriptions MPEG-7



Reference number
ISO/IEC TR 15938-8:2002(E)

AVA Documents
Document Sharing Hub
© ISO/IEC 2002

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/IEC 2002

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

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

Published in Switzerland

Contents

Foreword	vi
Introduction	vii
1 Scope.....	1
2 Terms and definitions	1
2.1 Conventions.....	1
2.1.1 Description tools	1
2.1.2 Naming convention	1
2.2 Terminology.....	2
2.2.1 Schema-related terminology	2
2.2.2 Content-related terminology	2
2.3 Symbols and abbreviated terms.....	6
2.3.1 Generic	6
2.3.2 Arithmetic operators	6
2.3.3 Logical operators.....	7
2.3.4 Relational operators.....	7
2.3.5 Bitwise operators.....	7
2.3.6 Conditional operators	7
2.3.7 Assignment	7
2.3.8 Constants	7
2.3.9 Functions.....	7
2.4 Default reference axis	8
3 MDS tools.....	8
3.1 Introduction	8
3.2 Schema tools.....	8
3.2.1 Introduction.....	8
3.2.2 Base types.....	8
3.2.3 Root element	8
3.2.4 Top-level types.....	10
3.2.5 Description metadata tools	18
3.3 Basic datatypes.....	21
3.3.1 Introduction.....	21
3.3.2 Integer datatypes	21
3.3.3 Real datatypes	21
3.3.4 Vectors and matrices	21
3.3.5 Probability datatypes	23
3.3.6 String datatypes.....	24
3.4 Linking, identification and localization tools	25
3.4.1 Introduction.....	25
3.4.2 References to Ds and DSs	25
3.4.3 Unique Identifier	26
3.4.4 Time description tools	26
3.4.5 Media Locators	29
3.5 Basic description tools.....	30
3.5.1 Introduction.....	30
3.5.2 Language identification	31
3.5.3 Textual annotation.....	32
3.5.4 Classification Schemes and Terms	37
3.5.5 Description of agents.....	49
3.5.6 Description of places	53
3.5.7 Graphs and relations.....	53
3.5.8 Ordering Tools.....	55
3.5.9 Affective description	56
3.5.10 Phonetic description	67
3.6 Media description tools	67
3.6.1 Introduction.....	67
3.6.2 Media information tools	68
3.7 Creation and production description tools	73

3.7.1	Introduction.....	73
3.7.2	Creation information tools.....	74
3.8	Usage description tools	76
3.8.1	Introduction.....	76
3.8.2	Usage information tools	77
3.9	Structure description tools	78
3.9.1	Introduction.....	78
3.9.2	Base segment description tools	79
3.9.3	Segment attribute description tools.....	80
3.9.4	Visual segment description tools	87
3.9.5	Audio segment description tools.....	109
3.9.6	Audio-visual segment description tools	110
3.9.7	Multimedia segment description tools.....	113
3.9.8	Ink segment description tools.....	114
3.9.9	Video editing segment description tools	122
3.9.10	Structural relation classification schemes	129
3.10	Semantics description tools	133
3.10.1	Introduction.....	133
3.10.2	Abstraction model	134
3.10.3	Semantic entity description tools	134
3.10.4	Semantic attribute description tools	150
3.10.5	Semantic relation classification schemes	153
3.11	Navigation and access tools.....	157
3.11.1	Introduction.....	157
3.11.2	Summarization	158
3.11.3	Views, partitions and decompositions.....	184
3.11.4	Variations of the content	199
3.12	Content organization tools.....	202
3.12.1	Introduction.....	202
3.12.2	Collections	202
3.12.3	Models	208
3.12.4	Probability models	209
3.12.5	Analytic models	214
3.12.6	Cluster models.....	219
3.12.7	Classification models.....	220
3.13	User interaction tools	223
3.13.1	Introduction.....	223
3.13.2	User preferences	223
3.13.3	Usage History.....	235
4	Visual tools	240
4.1	Basic visual tools	240
4.1.1	Grid layout.....	240
4.1.2	Visual time series	240
4.1.3	2D-3D multiple view.....	247
4.1.4	Spatial 2D coordinates.....	251
4.1.5	Temporal interpolation.....	254
4.2	Color description tools.....	257
4.2.1	Color space	257
4.2.2	Color quantization	258
4.2.3	Dominant color	259
4.2.4	Scalable color	262
4.2.5	Color layout.....	264
4.2.6	Color structure.....	268
4.2.7	GoF/GoP color	279
4.3	Texture description tools	280
4.3.1	Homogeneous texture.....	280
4.3.2	Texture browsing.....	283
4.3.3	Edge histogram.....	286
4.4	Shape description tools	291
4.4.1	Region-based shape	291

4.4.2	Contour-based shape.....	294
4.4.3	Shape 3D	298
4.5	Motion description tools	302
4.5.1	Camera motion.....	302
4.5.2	Motion trajectory.....	307
4.5.3	Parametric motion	309
4.5.4	Motion activity.....	313
4.6	Localization tools	319
4.6.1	Region locator.....	319
4.6.2	Spatio-temporal locator	322
4.7	Other visual tools	329
4.7.1	Face recognition	329
Annex A	Patent statements	338
Bibliography		340

Foreword

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

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

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

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC TR 15938-8, which is a Technical Report of type 3, 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*

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

1 Scope

This International Standard specifies a metadata system for describing multimedia content. This document gives examples of extraction and use of descriptions using Description Schemes, Descriptors, and datatypes specified in ISO/IEC 15938. The following set of subclauses are provided for each description tool, where optional subclauses are indicated as (optional):

- Informative examples (optional): provides informative examples that illustrate the instantiation of the description tool in creating descriptions.
- Extraction (optional): provides informative examples that illustrate the extraction of descriptions from multimedia content.
- Use (optional): provides informative examples that illustrate the use of descriptions.

This document is meant to be a companion technical report for Part 5 (Multimedia Description Schemes) and Part 3 (Visual) of ISO/IEC 15938. As such, the content of this technical report is not easily understood without the technical specifications. In this technical report, effort has been made to preserve the specific subclause numbering of ISO/IEC 15938-5 and ISO/IEC 15938-3 to allow easy correlation of the content on extraction and use in the technical report with the technical specifications.

2 Terms and definitions

2.1 Conventions

2.1.1 Description tools

This part of ISO/IEC 15938 specifies the multimedia description tools as follows:

- **Description Scheme (DS)** – a description tool that describes entities or relationships pertaining to multimedia content. DSs specify the structure and semantics of their components, which may be Description Schemes, Descriptors, or datatypes.
- **Descriptor (D)** – a description tool that describes a feature, attribute, or group of attributes of multimedia content.
- **Datatype** – a basic reusable datatype employed by Description Schemes and Descriptors.
- **Description Tool (or tool)** – refers to a Description Scheme, Descriptor, or Datatype.

2.1.2 Naming convention

In order to specify the multimedia description tools, this part of ISO/IEC 15938 uses constructs provided by the Description Definition Language (DDL) specified in ISO/IEC 15938-2, such as "element", "attribute", "simpleType" and "complexType". The names associated to these constructs are created on the basis of the following conventions:

- If the name is composed of multiple words, the first letter of each word is capitalized, with the exception that the capitalization of the first word depends on the type of construct as follows:
- Element naming: the first letter of the first word is capitalized (e.g. TimePoint element of TimeType).
- Attribute naming: the first letter of the first word is not capitalized (e.g. timeUnit attribute of IncrDurationType).
- complexType naming: the first letter of the first word is capitalized, and the suffix "Type" is used at the end of the name (e.g. PersonType).

- **simpleType** naming: the first letter of the first word is not capitalized, the suffix "Type" may be used at the end of the name (e.g. `timePointType`).

Note that when referencing a **complexType** or **simpleType** in the definition of a description tool, the "Type" suffix is not used. For instance, the text refers to the "Time datatype" (instead of "TimeType datatype"), to the "MediaLocator D" (instead of "MediaLocatorType D") and to the "Person DS" (instead of "PersonType DS").

2.2 Terminology

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

2.2.1 Schema-related terminology

2.2.1.1

Attribute

A field of a **description tool** which is of simple type.

2.2.1.2

Base type

A **type** that serves as the root **type** of a derivation hierarchy for other **types**.

2.2.1.3

Datatype

A primitive reusable **type** employed by **Description Schemes** and **Descriptors**.

2.2.1.4

Derived type

A **type** that is defined in terms of extension or restriction of other **types**.

2.2.1.5

Description

An instantiation of one or more **description tools**.

2.2.1.6

Description Scheme

A **description tool** that describes 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**.

2.2.1.7

Description Tool

A **Description Scheme**, **Descriptor**, or **datatype**.

2.2.1.8

Descriptor

A description tool that describes a feature, **attribute**, or group of attributes of multimedia content.

2.2.1.9

Instantiation

Assignment of values to the fields (elements, attributes) of one or more **description tools**.

2.2.1.10

Element

A field of a **description tool** which is of complex type.

2.2.1.11

Schema

The set of related **description tools**, for example, those specified in ISO/IEC 15938.

2.2.1.12

Type

The format used for collection of letters, digits, and/or symbols, to depict values of an element or attribute of **description tool**. A **type** consists of a set of distinct values, a set of lexical representations, and a set of facets that characterize properties of the value space, individual values, or lexical items.

2.2.2 Content-related terminology

2.2.2.1

Abstraction

A secondary representation that is created from or is related to the **content**. For example, a **summary** of a **video** or a **model** of a **feature**.

2.2.2.2**AC coefficient**

Any DCT coefficient for which the frequency in one or both dimensions is non-zero.

2.2.2.3**Acquisition**

The process of acquiring **audio** or **visual** data from a source.

2.2.2.4**Action**

A semantically identifiable behavior of an object or group of objects, for example, a soccer player kicking ball.

2.2.2.5**Agent**

A person, organization, or group of persons.

2.2.2.6**Audio**

Time-varying **data** or signal intended for listening or hearing. Also, related to the aural modality.

2.2.2.7**Audio-visual**

content consisting of both **audio** and **video** data.

2.2.2.8**Automatic**

Processing of **multimedia data**, **content**, or **metadata** by means of computer, hardware, or other software device.

2.2.2.9**Classification Scheme**

A list of defined terms and their meanings.

2.2.2.10**Content****Multimedia content**

A representation of the information contained in or related to **multimedia data** in a formalized manner suitable for interpretation by human means. **Content** refers to the **data** and the **metadata**.

2.2.2.11**Copyright**

A right that establishes the ownership of **data**, **content**, or **metadata**.

2.2.2.12**Data****Essence****Multimedia Data**

A representation of **multimedia** in a formalized manner suitable for communication, interpretation, or processing by automatic means.

2.2.2.13**DC coefficient**

The DCT coefficient for which the frequency in both dimensions is zero.

2.2.2.14**DCT coefficient**

The signed amplitude of a specific cosine basis function.

2.2.2.15**Editing**

The process of combining, extracting, and refining **multimedia data**.

2.2.2.16**Eigenface**

An eigenvector obtained from the principal component analysis of facial images.

2.2.2.17**Entity**

Any concrete or abstract thing of interest related to the **multimedia content**.

2.2.2.18

Event

A noteworthy occurrence that happens at a point in time or during a temporal interval. Alternatively used as a change in state.

2.2.2.19

Feature

A distinctive characteristic of **multimedia content** that signifies something to a human observer, such as the "color" or "texture" of an image.

2.2.2.20

Filtering

A process for selecting multimedia content that satisfies certain criteria. This process may include ranking the content according to the extent that it satisfies the criteria.

2.2.2.21

Format

The characteristics of the stored or physical representation of the **data**.

2.2.2.22

Frame

A single **image** from a **video**.

2.2.2.23

Image

2D spatially-varying visual data acquired from a visual source.

2.2.2.24

Key frame

A representative **frame** of a **video** or a **segment**.

2.2.2.25

Locator

Specifies the location or address of **multimedia data** or a **segment**.

2.2.2.26

Model

A parametric or statistical representation of multimedia **content** or **features**.

2.2.2.27

Manual

Processing of **multimedia data**, **content**, or **metadata** by human means.

2.2.2.28

Metadata

The information and documentation which makes **multimedia data** understandable and shareable to users over time.

2.2.2.29

Multimedia

Data comprising one or modalities, such as images, audio, video, 3D models, ink content, and so forth.

2.2.2.30

Navigation

A process by which a **user** accesses **multimedia content** and steers a course through the content in a controlled manner.

2.2.2.31

Object

An object with a physical representation in the natural world.

2.2.2.32

Region

A spatial unit of **multimedia**, for example, a 2D spatial **region** of an **image**, or a moving region of **video**.

2.2.2.33

Relation

Any association among entities.

2.2.2.34**Rights**

Information that determines the ownership and terms of use of **multimedia data, content, or metadata**. Refers to Intellectual Property Rights, Copyrights, and the Access Rights.

2.2.2.35**Scene**

An episode or sequence of events representing continuous action in one location.

2.2.2.36**Search**

A process for searching **multimedia content** that satisfies certain criteria. This process may include ranking the content according to the extent that it satisfies the criteria.

2.2.2.37**Segment**

A spatial or temporal unit of **multimedia**, for example, a temporal **segment of video**, or a **segment of an image**.

2.2.2.38**Semantics**

Information relating to the underlying meaning or understanding of **multimedia content**. Alternatively, refers to the specification of the meaning of **description tools**.

2.2.2.39**Summary**

An abstraction of **multimedia content** that summarizes the **content**.

2.2.2.40**User**

An end-user or consumer of **multimedia content**.

2.2.2.41**User Preferences**

The preferences of a **user** pertaining to **multimedia content**. This includes the user's tastes, likes and dislikes with respect to the content and its properties, as well as preferences with respect to the consumption process.

2.2.2.42**Usage History**

A history of actions that a **user of multimedia content** has carried out over a certain period of time, such as recording a specific piece of content, or playing back recorded content at a specific time.

2.2.2.43**Variation**

An alternative version of **multimedia content**, which may be derived through transcoding, summarization, translation, reduction, and so forth.

2.2.2.44**Video**

A space- and time-varying visual **data** or **signal** intended for viewing; commonly represented as a discrete sequence of **images** or **frames**.

2.2.2.45**View**

A portion of an **image, video** or **audio** signal, defined in terms of a partition. A partition is a multi-dimensional region defined in the space, time and/or frequency plane.

2.2.2.46**Visual**

Related to the visual modality.

2.2.2.47**View Decomposition**

An organized set of **views** that provides a structured decomposition of an **image, video** or **audio** signal in multi-dimensional space, time and/or frequency.

2.2.2.48**3D mesh model**

Representation model of the surface of 3D objects using a set of faces and nodes. (i.e. polygonal meshes)

2.3 Symbols and abbreviated terms

2.3.1 Generic

For the purposes of this part of ISO/IEC 15938, the symbols and abbreviated terms given in the following apply:

ART:	Angular-Radial Transform
AV:	Audio-visual
CSS:	Curvature Scale Space
CIE:	International Commission on Illumination
CIF:	Common Intermediate Format
CS:	Classification Scheme
D:	Descriptor
Ds:	Descriptors
DCT:	Discrete Cosine Transform
DDL:	Description Definition Language
DS:	Description Scheme
DSs:	Description Schemes
FOC:	Focus of Contraction
FOE:	Focus of Expansion
GLA:	Generalized Lloyd Algorithm
GoF:	Group of Frames
GoP:	Group of Pictures
HMMD:	Hue-Min-Max-Difference
HSV:	Hue-Saturation-Value
IANA:	Internet Assigned Numbers Authority
IETF:	Internet Engineering Task Force
IPMP:	Intellectual Property Management and Protection
ISO:	International Organization for Standardization
JPEG:	Joint Photographic Experts Group
MDS:	Multimedia Description Scheme
MNV:	Mean Normal Vector
MPEG:	Moving Picture Experts Group
MPEG-2:	Generic coding of moving pictures and associated audio information (see ISO/IEC 13818)
MPEG-4:	Coding of audio-visual objects (see ISO/IEC 14496)
MPEG-7:	Multimedia Content Description Interface Standard (see ISO/IEC 15938)
MP3:	MPEG-2 layer 3 audio coding
NAC:	Normalized Auto-Correlation
QCIF:	Quarter Common Intermediate Format
PWM:	Pseudo Weighted Measure
RGB:	Red-Green-Blue
SMPTE:	Society of Motion Picture and Television Engineers
SSD:	Shape Spectrum Descriptor
TZ:	Time Zone
TZD:	Time Zone Difference
URI:	Uniform Resource Identifier (see RFC 2396)
URL:	Uniform Resource Locator (see RFC 2396)
W3C:	World Wide Web Consortium
XML:	Extensible Markup Language
XOR :	eXclusive-OR

2.3.2 Arithmetic operators

+	Addition
-	Subtraction (as a binary operator) or negation (as a unary operator)
++	Increment, i.e. $x++$ is equivalent to $x=x+1$
--	Decrement, i.e. $x--$ is equivalent to $x=x-1$
+=	Accumulation, i.e. $x+=2$ is equivalent to $x=x+2$
/=	divide and substitute, i.e. $x/=2$ is equivalent to $x=x/2$
*	Multiplication
x	Multiplication
^	Power

- / Integer division with truncation of the result towards zero. For example, 7/4 and -7/-4 are truncated to 1, -7/4 and 7/-4 are truncated to -1.
- // Integer division with rounding to the nearest integer. Half-integer values are rounded away from zero unless otherwise specified. For example, 3//2 is rounded to 2, and -3//2 is rounded to -2.
- ÷ Used to indicate division in mathematical equations where no rounding is intended
- % Modulus operator, defined only for positive numbers
- ld Logarithm base 2
- ceil Minimum integer number greater or equal than the given floating point number

$$\text{Sign() } \text{Sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

$$\text{Abs() } \text{Abs}(x) = \begin{cases} x & x \geq 0 \\ -x & x < 0 \end{cases}$$

$$\sum_{i=a}^{i=b} f(i) \text{ Summation of } f(i) \text{ with } i \text{ taking integer values from } a \text{ up to, but not including } b.$$

2.3.3 Logical operators

- || Logical OR
- && Logical AND
- ! Logical NOT

2.3.4 Relational operators

- > Greater than
- >= Greater than or equal to
- ≥ Greater than or equal to
- < Less than
- <= Less than or equal to
- ≤ Less than or equal to
- == Equal to
- != Not equal to
- max[] Maximum value in argument list
- min[] Minimum value in argument list
- median[] median value in argument list

2.3.5 Bitwise operators

- | OR
- & AND
- >> Shift right with sign extension
- << Shift left with zero fill

2.3.6 Conditional operators

$$?: \text{ condition} ? a : b = \begin{cases} a & \text{if (condition)} \\ b & \text{otherwise} \end{cases}$$

2.3.7 Assignment

- = Assignment operator

2.3.8 Constants

- π 3.141 592 653 58...
- e 2.718 281 828 45...

2.3.9 Functions

- max() Maximum value in argument list
- min() Minimum value in argument list

$$\text{Sign() } \text{Sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

$$\text{Abs() } \text{Abs}(x) = \begin{cases} x & x \geq 0 \\ -x & x < 0 \end{cases}$$

$$\sum_{i=a}^{i=b} f(i) \text{ Summation of } f(i) \text{ with } i \text{ taking integer values from } a \text{ up to, but not including } b.$$

Distances between N-dimensional vectors **x** and **y**

$$\text{L1 norm } L1(\mathbf{x}, \mathbf{y}) = \sum_i^N |x_i - y_i|$$

$$\text{L2 norm } L2(\mathbf{x}, \mathbf{y}) = \sum_i (x_i - y_i)^2$$

$$\text{Euclidean distance } D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

2.4 Default reference axis

The default reference axis for angle calculation is the positive *x* (horizontal) axis. Positive angle is calculated anti-clockwise.

3 MDS tools

3.1 Introduction

Clause 3.1 provides guidelines and examples of the extraction and use of descriptions for tools defined in ISO/IEC 15938-5.

3.2 Schema tools

3.2.1 Introduction

This clause specifies the schema tools that facilitate the making of descriptions. The following description tools are specified: (1) the base type hierarchy of the description tools defined in ISO/IEC 15938, (2) the root element, (3) the top-level tools, (4) the multimedia content entity tools, (5) package tool, and (6) description metadata tool. The functionality of these tools is given as follows:

Table 1 - Overview of Schema Tools.

<i>Tool</i>	<i>Functionality</i>
Base types	Form the type hierarchy for description tools.
Root element	The initial wrapper or root element of descriptions.
Top-level tools	The elements that follow the root element in descriptions.
Multimedia content entity tools	Tools for describing different types of multimedia content such as images, video, audio, mixed multimedia, collections, and so forth.
Package	Tool for organizing or packaging of description tools.
Description metadata	Tool for describing metadata about descriptions.

3.2.2 Base types

No additional informative material for extraction and use is provided.

3.2.3 Root element

3.2.3.1 Root element examples

The following example shows the use of the root element for describing an instance of the ScalableColor D (defined in ISO/IEC 15398-3) using DescriptionUnit.

```

<Mpeg7>
  <DescriptionMetadata>
    <Version>1.0</Version>
    <PrivateIdentifier>descriptionUnitExample</PrivateIdentifier>
  </DescriptionMetadata>
  <DescriptionUnit xsi:type="ScalableColorType" numOfCoeff="16"
    numOfBitplanesDiscarded="0">
    <Coeff> 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 </Coeff>
  </DescriptionUnit>
</Mpeg7>

```

The following example shows the use of root element for describing an image using Description.

```

<Mpeg7>
  <DescriptionMetadata>
    <Confidence>1.0</Confidence>
    <Version>1.1</Version>
    <LastUpdate>2001-09-20T03:20:25+09:00</LastUpdate>
    <PublicIdentifier>
      <IDOrganization>
        <Name>ISO</Name>
      </IDOrganization>
      <IDName>
        <Name>International Organization of Standardization</Name>
      </IDName>
      <UniqueID>098f2470-bae0-11cd-b579-08002b30bfeb</UniqueID>
    </PublicIdentifier>
    <PrivateIdentifier>completeDescriptionExample</PrivateIdentifier>
    <Creator>
      <Role href="creatorCS"><Name>Creator</Name><Role>
      <Person>
        <Name>
          <GivenName>Yoshiaki</GivenName>
          <FamilyName>Shibata</FamilyName>
        </Name>
      </Person>
    </Creator>
    <CreationLocation>
      <Country>jp</Country>
      <AdministrativeUnit>Tokyo</AdministrativeUnit>
    </CreationLocation>
    <CreationTime>2000-10-10T19:45:00+09:00</CreationTime>
    <Instrument>
      <Tool>
        <Name>Wizzo Extracto ver. 2</Name>
      </Tool>
      <Setting name="sensitivity" value="0.5"/>
    </Instrument>
    <Rights> RID# </Rights>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <!-- more elements here -->
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

3.2.4 Top-level types

3.2.4.1 Complete description types

3.2.4.1.1 Content description types examples

The following example shows the use of the content description type `ContentEntityType` for describing a photographic image depicting a sunset. The image description includes a text annotation and a description of the image color using `ScalableColor D` (defined in ISO/IEC 15938-3).

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaLocator>
          <MediaUri>image.jpg</MediaUri>
        </MediaLocator>
        <TextAnnotation>
          <FreeTextAnnotation> Sunset scene </FreeTextAnnotation>
        </TextAnnotation>
        <VisualDescriptor xsi:type="ScalableColorType" numOfCoeff="16"
          numOfBitplanesDiscarded="0">
          <Coeff> 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 </Coeff>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

The following example shows the use of the content description type `SemanticDescriptionType` for describing a car that is depicted in an image (image.jpg). The `Semantic DS` is defined in 3.10.3.3.

```
<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Car </Name>
      </Label>
      <Definition>
        <FreeTextAnnotation>
          Four wheel motorized vehicle
        </FreeTextAnnotation>
      </Definition>
      <MediaOccurrence>
        <MediaLocator>
          <MediaUri> image.jpg </MediaUri>
        </MediaLocator>
      </MediaOccurrence>
    </Semantics>
  </Description>
</Mpeg7>
```

The following example shows the use of the content description type `ModelDescriptionType` for describing a collection model of "sunsets" that contains two images depicting sunset scenes. The `CollectionModel DS` is defined in 3.12.5.2.

```
<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="CollectionModelType" confidence="0.75" reliability="0.5"
      function="described">
      <Label>
        <Name>Sunsets</Name>
      </Label>
    <Collection xsi:type="ContentCollectionType">
```

```

        <Content xsi:type="ImageType">
            <Image>
                <MediaLocator xsi:type="ImageLocatorType">
                    <MediaUri>sunset1.jpg</MediaUri>
                </MediaLocator>
            </Image>
        </Content>
        <Content xsi:type="ImageType">
            <Image>
                <MediaLocator xsi:type="ImageLocatorType">
                    <MediaUri>sunset2.jpg</MediaUri>
                </MediaLocator>
            </Image>
        </Content>
    </Collection>
</Model>
</Description>
</Mpeg7>

```

The following example shows the use of the content description type `SummaryDescriptionType` for describing a hierarchical summary of video. The hierarchical summary has two levels: level 0 consists of three summary segments, and level 1 consists of one summary segment. The `HierarchicalSummary` DS is defined in 3.11.2.3.

```

<Mpeg7>
    <Description xsi:type="SummaryDescriptionType">
        <Summarization>
            <Summary xsi:type="HierarchicalSummaryType"
                components="keyVideoClips" hierarchy="dependent">
                <SourceLocator>
                    <MediaUri>file://disk/video001.mpg</MediaUri>
                </SourceLocator>
                <SummarySegmentGroup level="0">
                    <SummarySegment> <!-- segment #1 at level 0 -->
</SummarySegment>
                    <SummarySegment> <!-- segment #2 at level 0 -->
</SummarySegment>
                    <SummarySegment> <!-- segment #3 at level 0 -->
</SummarySegment>
                    <SummarySegmentGroup level="1">
                        <SummarySegment> <!-- segment #1 at level 1 -->
</SummarySegment>
                    </SummarySegmentGroup>
                </SummarySegmentGroup>
            </Summary>
        </Summarization>
    </Description>
</Mpeg7>

```

The following example shows the use of the content description type `ViewDescriptionType` for describing a spatial view of an image, which corresponds to the upper-right quarter of the image. The `SpaceView` DS is defined in 3.11.3.5.

```

<Mpeg7>
    <Description xsi:type="ViewDescriptionType">
        <View xsi:type="SpaceViewType">
            <Target>
                <ImageSignal>
                    <MediaLocator>
                        <MediaUri> view.jpg </MediaUri>
                    </MediaLocator>
                </ImageSignal>
            </Target>

```

```

    <Source>
      <ImageSignal>
        <MediaLocator>
          <MediaUri> image.jpg </MediaUri>
        </MediaLocator>
      </ImageSignal>
    </Source>
    <SpacePartition>
      <Origin xOrigin="left" yOrigin="top"/>
      <Start xsi:type="SignalPlaneFractionType" x="0.5" y="0"/>
      <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
    </SpacePartition>
  </View>
</Description>
</Mpeg7>

```

The following example shows the use of the content description type `VariationDescriptionType` for describing an image that is a variation of a video in which the image has been extracted from the video. The `VariationSet DS` is defined in 3.11.4.1.

```

<Mpeg7>
  <Description xsi:type="VariationDescriptionType">
    <VariationSet>
      <Source xsi:type="mpeg7:VideoType">
        <Video>
          <MediaLocator>
            <MediaUri>file://video-A.mpg</MediaUri>
          </MediaLocator>
        </Video>
      </Source>
      <Variation fidelity="0.85" priority="1">
        <Content xsi:type="ImageType">
          <Image>
            <MediaLocator>
              <MediaUri>file://image-B.jpg</MediaUri>
            </MediaLocator>
          </Image>
        </Content>
        <VariationRelationship>extraction</VariationRelationship>
      </Variation>
    </VariationSet>
  </Description>
</Mpeg7>

```

3.2.4.2 Content management types

3.2.4.2.1 Content management types example

The following example shows the use of the content management type `UserDescriptionType` for describing the user preferences of a user of a multimedia system. The `UserPreferences DS` is defined in 3.13.2.1.

```

<Mpeg7>
  <Description xsi:type="UserDescriptionType">
    <User xsi:type="PersonType">
      <Name xml:lang="en">
        <GivenName> John </GivenName>
        <FamilyName> Smith </FamilyName>
      </Name>
    </User>
    <UserPreferences>
      <UserIdentifier>
        <Name> jrsmith </Name>
      </UserIdentifier>
    </UserPreferences>
  </Description>
</Mpeg7>

```

```

        </UserIdentifier>
        <FilteringAndSearchPreferences>
            <!-- more elements here -->
        </FilteringAndSearchPreferences>
    </UserPreferences>
</Description>
</Mpeg7>

```

The following example shows the use of the content management type `MediaDescriptionType` for describing the media information of a news video. The `MediaInformation` DS is defined in 3.6.2.1.

```

<Mpeg7>
  <Description xsi:type="MediaDescriptionType">
    <MediaInformation id="news1_media">
      <MediaIdentification>
        <EntityIdentifier organization="MPEG" type="MPEG7ContentSetId">
          mpeg7_content:news1
        </EntityIdentifier>
        <VideoDomain href="urn:mpeg:mpeg7:cs:VideoDomainCS:2001:1.2">
          <Name xml:lang="en">Natural</Name>
        </VideoDomain>
      </MediaIdentification>
      <MediaProfile>
        <MediaFormat>
          <Content href="MPEG7ContentCS">
            <Name>audiovisual</Name>
          </Content>
          <Medium href="urn:mpeg:mpeg7:cs:MediumCS:2001:1.1">
            <Name xml:lang="en">CD</Name>
          </Medium>
          <FileFormat href="urn:mpeg:mpeg7:cs:FileFormatCS:2001:3">
            <Name xml:lang="en">mpeg</Name>
          </FileFormat>
          <FileSize>666478608</FileSize>
          <VisualCoding>
            <Format href="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:1"
              colorDomain="color">
              <Name xml:lang="en">MPEG-1 Video</Name>
            </Format>
            <Pixel aspectRatio="0.75" bitsPer="8"/>
            <Frame height="288" width="352" rate="25"/>
          </VisualCoding>
        </MediaFormat>
      </MediaProfile>
    </MediaInformation>
  </Description>
</Mpeg7>

```

The following example shows the use of the content management type `CreationDescriptionType` for describing the creation information for a sports video. The `CreationInformation` DS is defined in 3.7.2.1.

```

<Mpeg7>
  <Description xsi:type="CreationDescriptionType">
    <CreationInformation>
      <Creation>
        <Title xml:lang="en" type="popular">Subway series</Title>
        <Abstract>
          <FreeTextAnnotation> Game among city rivals
          </FreeTextAnnotation>
        </Abstract>
        <Creator>
          <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:PUBLISHER"/>

```

```

        <Agent xsi:type="OrganizationType">
            <Name>Sports Channel</Name>
        </Agent>
    </Creator>
</Creation>
</CreationInformation>
</Description>
</Mpeg7>

```

The following example shows the use of the content management type UsageDescriptionType for describing the usage information for a sport video released for the Internet. The UsageInformation DS is defined in 3.8.2.1.

```

<Mpeg7>
  <Description xsi:type="UsageDescriptionType">
    <UsageInformation>
      <Rights>
        <RightsID organization="NBC" type="NBCCopyRightsID">
          nbc:20010618:td2
        </RightsID>
      </Rights>
      <Availability id="sports">
        <InstanceRef href="onlinemp7cs17sports1"/>
        <Dissemination>
          <Format href="urn:mpeg:mpeg7:cs:DisseminationFormatCS:2001:4">
            <Name xml:lang="en">Internet</Name>
          </Format>
          <Location>
            <Region>us</Region>
          </Location>
        </Dissemination>
        <AvailabilityPeriod type="payPerUse">
          <TimePoint>2001-06-16T21:00+01:00</TimePoint>
          <Duration>PT30M</Duration>
        </AvailabilityPeriod>
      </Availability>
    </UsageInformation>
  </Description>
</Mpeg7>

```

The following example shows the use of the content management type ClassificationSchemeDescriptionType for describing a classification scheme for audio domain. The ClassificationScheme DS is defined in 3.5.4.1.

```

<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationScheme uri="urn:mpeg:mpeg7:cs:MyAudioDomainCS"
      domain="//MediaInformation/MediaProfile/MediaIdentification/AudioDomain">
      <Header xsi:type="DescriptionMetadataType">
        <Version>1.0</Version>
      </Header>
      <Term termID="1">
        <Name xml:lang="en">Source</Name>
        <Definition xml:lang="en">Type of audio source</Definition>
        <Term termID="1.1">
          <Name xml:lang="en">Synthetic</Name>
        </Term>
        <Term termID="1.2">
          <Name xml:lang="en">Natural</Name>
        </Term>
      </Term>
    </ClassificationScheme>
  </Description>
</Mpeg7>

```

```

    </Description>
  </Mpeg7>

```

3.2.4.3 Multimedia content entity description tools

3.2.4.3.1 Multimedia content entity description tools examples

The following example shows the use of the multimedia content entity ImageType for describing an image.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaLocator>
          <MediaUri>image.jpg</MediaUri>
        </MediaLocator>
        <CreationInformation>
          <Creation>
            <Title> World Series Game 3 </Title>
          </Creation>
        </CreationInformation>
        <TextAnnotation>
          <FreeTextAnnotation> Game winning homerun </FreeTextAnnotation>
        </TextAnnotation>
        <Semantic>
          <Label>
            <Name>Last inning</Name>
          </Label>
          <SemanticBase xsi:type="EventType">
            <Label>
              <Name>Game winning homerun </Name>
            </Label>
          </SemanticBase>
        </Semantic>
        <VisualDescriptor xsi:type="ScalableColorType" numOfCoeff="16"
          numOfBitplanesDiscarded="0">
          <Coeff> 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 </Coeff>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

The following example shows the use of the multimedia content entity VideoType for describing a video.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <CreationInformation>
          <Creation>
            <Title> Worldcup Soccer </Title>
          </Creation>
        </CreationInformation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT1M30S</MediaDuration>
        </MediaTime>
        <VisualDescriptor xsi:type="GoFGoPColorType"
          aggregation="Average">
          <ScalableColor numOfCoeff="16" numOfBitplanesDiscarded="0">
            <Coeff> 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 </Coeff>
          </ScalableColor>
        </VisualDescriptor>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

```

    </Video>
  </MultimediaContent>
</Description>
</Mpeg7>

```

The following example shows the use of the multimedia content entity AudioType for describing audio content.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioType">
      <Audio>
        <CreationInformation>
          <Creation>
            <Title> Morning Radio </Title>
          </Creation>
        </CreationInformation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT30M00S</MediaDuration>
        </MediaTime>
      </Audio>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

The following example shows the use of the multimedia content entity AudioVisualType for describing audiovisual content.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioVisualType">
      <AudioVisual>
        <MediaLocator>
          <MediaUri>program.mpg</MediaUri>
        </MediaLocator>
        <CreationInformation>
          <Creation>
            <Title> News Flash</Title>
          </Creation>
        </CreationInformation>
        <TextAnnotation>
          <FreeTextAnnotation> Afternoon news program
</FreeTextAnnotation>
        </TextAnnotation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT10M300S</MediaDuration>
        </MediaTime>
      </AudioVisual>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

The following example shows the use of the multimedia content entity MultimediaType for describing multimedia content.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaType">
      <Multimedia>
        <MediaSourceDecomposition gap="false" overlap="false">
          <Segment xsi:type="StillRegionType">

```

```

        <TextAnnotation>
          <FreeTextAnnotation> image </FreeTextAnnotation>
        </TextAnnotation>
      </Segment>
      <Segment xsi:type="VideoSegmentType">
        <TextAnnotation>
          <FreeTextAnnotation> video </FreeTextAnnotation>
        </TextAnnotation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT0M15S</MediaDuration>
        </MediaTime>
      </Segment>
      <Segment xsi:type="AudioSegmentType">
        <TextAnnotation>
          <FreeTextAnnotation> audio </FreeTextAnnotation>
        </TextAnnotation>
      </Segment>
    </MediaSourceDecomposition>
  </Multimedia>
</MultimediaContent>
</Description>
</Mpeg7>

```

The following example shows the use of the multimedia content entity `MultimediaCollectionType` for describing a collection consisting of one image and one video.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <Collection xsi:type="ContentCollectionType">
        <Content xsi:type="ImageType">
          <Image>
            <MediaLocator xsi:type="ImageLocatorType">
              <MediaUri>image.jpg</MediaUri>
            </MediaLocator>
          </Image>
        </Content>
        <Content xsi:type="VideoType">
          <Video>
            <MediaLocator xsi:type="TemporalSegmentLocatorType">
              <MediaUri>video.mpg</MediaUri>
            </MediaLocator>
          </Video>
        </Content>
      </Collection>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

The following example shows the use of the multimedia content entity `SignalType` for describing an image signal.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="SignalType">
      <ImageSignal>
        <MediaLocator xsi:type="ImageLocatorType">
          <MediaUri>video.mpg</MediaUri>
        </MediaLocator>
      </ImageSignal>
    </MultimediaContent>
  </Description>

```

```
</Mpeg7>
```

The following example shows the use of the multimedia content entity InkContentType for describing ink content.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="InkContentType">
      <InkContent>
        <MediaLocator>
          <MediaUri>ink.mpg</MediaUri>
        </MediaLocator>
        <InkMediaInformation>
          <InputDevice resolutionS="133">
            <Device>
              <Name xml:lang="en">PenDevice</Name>
            </Device>
          </InputDevice>
          <Handedness>right</Handedness>
          <Style>mixed</Style>
        </InkMediaInformation>
      </InkContent>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

The following example shows the use of the multimedia content entity AnalyticEditedVideoType for describing an edited video.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AnalyticEditedVideoType">
      <AnalyticEditedVideo xsi:type="EditedVideoType">
        <MediaLocator>
          <MediaUri> video.mpg </MediaUri>
        </MediaLocator>
        <CreationInformation>
          <Creation>
            <Title> Worldcup Soccer </Title>
          </Creation>
        </CreationInformation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT1M30S</MediaDuration>
        </MediaTime>
      </AnalyticEditedVideo>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

3.2.5 Description metadata tools

3.2.5.1 Package DS

3.2.5.1.1 Package DS examples

The following example shows the use of the Package DS for describing a package of description tools for collections and models. The package consists of one package named "Content Organization", which consists of another package that lists four tools: ContentCollectionType, DescriptorCollectionType, ConceptCollectionType, and MixedCollectionType, and another package named "Probability Models", which consists of two additional packages named: "Probability Distributions", "Continuous Probability Distribution", and "Finite State Models"

```

<Mpeg7>
  <DescriptionMetadata>
    <Version>1.0</Version>
    <Package name="Content Organization">
      <Package name="Collections">
        <Scheme name="ContentCollectionType"/>
        <Scheme name="DescriptorCollectionType"/>
        <Scheme name="ConceptCollectionType"/>
        <Scheme name="MixedCollectionType"/>
      </Package>
      <Package name="Models">
        <Package name="Probability Models">
          <Package name="Probability Distributions">
            <Package name="Discrete Probability Distributions">
              <Scheme name="HistogramProbabilityType"/>
              <Scheme name="BinomialDistributionType"/>
              <Scheme name="PoissonDistributionType"/>
            </Package>
            <Package name="Continuous Probability Distributions">
              <Scheme name="GaussianDistributionType"/>
              <Scheme name="GeneralizedGaussianDistributionType"/>
            </Package>
          </Package>
          <Package name="Finite State Models">
            <Scheme name="StateTransitionModelType"/>
            <Scheme name="HiddenMarkovModelType"/>
          </Package>
        </Package>
      </Package>
    </DescriptionMetadata>
    <!-- more elements here -->
  </Mpeg7>

```

3.2.5.2 DescriptionMetadata Header

3.2.5.2.1 DescriptionMetadata Header examples

The following example shows the use of the DescriptionMetadata Header for describing a video segment. Note that this example shows how description metadata can be attached to one component of the description — in this case the segment contained within the video segment temporal decomposition.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <MediaLocator>
          <MediaUri>video.mpg</MediaUri>
        </MediaLocator>
        <TextAnnotation>
          <FreeTextAnnotation> Soccer scene </FreeTextAnnotation>
        </TextAnnotation>
        <TemporalDecomposition gap="false" overlap="false"
          decompositionType="temporal">
          <VideoSegment xsi:type="VideoSegmentType">
            <Header xsi:type="DescriptionMetadataType">
              <Confidence>0.75</Confidence>
              <Version>1.4</Version>
              <LastUpdate>2001-03-11T10:00:00+00:00</LastUpdate>
              <Comment>
                <FreeTextAnnotation>Extracted video segment
              </FreeTextAnnotation>
            </Comment>
            <PublicIdentifier type="UUID" organization="ISO">

```

```

0f93sjd8-h38f-20dk-kjf9-02kj78li
</PublicIdentifier>
<PrivateIdentifier>Monster Jr.</PrivateIdentifier>
<Creator>
  <Role href="creatorCS">
    <Name>Creator</Name>
  </Role>
  <Agent xsi:type="OrganizationType">
    <Name>MPEG-7 MDS Group</Name>
    <Kind>
      <Name>ISO MPEG subgroup</Name>
    </Kind>
    <Contact xsi:type="PersonType">
      <Name>
        <GivenName>John</GivenName>
        <FamilyName>Smith</FamilyName>
      </Name>
      <ElectronicAddress>
        <Email>nobody@nowhere.com</Email>
      </ElectronicAddress>
    </Contact>
    <ElectronicAddress>
      <Url>http://www.cselt.it/mpeg/</Url>
    </ElectronicAddress>
  </Agent>
</Creator>
<CreationLocation>
  <Region>en</Region>
  <AdministrativeUnit>New York</AdministrativeUnit>
</CreationLocation>
<CreationTime>2001-02-20T10:00:00+00:00</CreationTime>
<Instrument>
  <Tool>
    <Name>MPEG-7 WizzoExtracto Tool</Name>
  </Tool>
  <Setting name="precision" value="highest"/>
</Instrument>
<Rights>
  <RightsID organization="MPEG" type="MPEGcopyright">
    mpeg7:98shdj28:021
  </RightsID>
</Rights>
<Package name="Sports - Soccer Package">
  <Scheme name="GraphType"/>
  <Scheme name="StructuredAnnotationType"/>
  <Scheme name="PointOfViewType"/>
  <Scheme name="VideoSegmentType"/>
  <Scheme name="SummarizationType"/>
  <Scheme name="ViewType"/>
  <Scheme name="SemanticsType"/>
  <Scheme name="UserPreferencesType"/>
</Package>
</Header>
<!-- more elements here -->
</VideoSegment>
<!-- more elements here -->
</TemporalDecomposition>
<!-- more elements here -->
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

3.3 Basic datatypes

3.3.1 Introduction

This clause specifies a set of basic datatypes that are used to by the ISO/IEC 15938 description tools. While the DDL provides some built-in datatypes, additional datatypes required for the description of multimedia content are defined in this clause:

Table 2 - Overview of Basic Datatypes.

<i>Tool</i>	<i>Functionality</i>
Integer & Real datatypes	Tools for representing constrained integer and real values. A set of unsigned integer datatypes "unsignedXX" (where XX is the number of bits in the representation) represent integer values from 1 to 32 bits in length. In addition, several different datatypes representing constrained ranges of real values are specified: <code>minusOneToOne</code> , <code>zeroToOne</code> , and so on.
Vectors & Matrix datatypes	Tools for representing arbitrary sized vectors and matrices of integer or real values. For vectors, the <code>IntegerVector</code> , <code>FloatVector</code> , and <code>DoubleVector</code> datatypes represent vectors of integer, float, and double values respectively. For matrices, the <code>IntegerMatrix</code> , <code>FloatMatrix</code> , and <code>DoubleMatrix</code> datatypes represent matrices of integer, float, and double values respectively.
Probability Vectors & Matrix datatypes	Tools for representing probability distributions using vectors (<code>ProbabilityVector</code>) and matrices (<code>ProbabilityMatrix</code>).
String Datatypes	Tools for identifying content type (<code>mimeType</code>), countries (<code>countryCode</code>), regions (<code>regionCode</code>), currencies (<code>currencyCode</code>), and character sets (<code>characterSetCode</code>).

3.3.2 Integer datatypes

3.3.2.1 Unsigned datatypes

Information on extraction and use is not provided.

3.3.3 Real datatypes

3.3.3.1 zeroToOne datatype

Information on extraction and use is not provided.

3.3.3.2 minusOneToOne datatype

Information on extraction and use is not provided.

3.3.3.3 nonNegativeReal datatype

Information on extraction and use is not provided.

3.3.4 Vectors and matrices

3.3.4.1 Representing vector and matrix values

The datatypes representing vectors (`integerVector`, `floatVector`, `doubleVector`) and the datatypes representing matrix values (`IntegerMatrixType`, `FloatMatrixType`, `DoubleMatrix`) follow the specification found in ISO/IEC 15938 Part 2 (DDL) (see 5.1). The following is a brief summary of that subclause, which is needed to understand how the vector and matrix datatypes are defined and used.

In this standard, vectors and matrices differ in several respects. First, matrices are more general and can have any dimension, of one or higher. On the other hand vectors are a special kind of matrix that is limited to being one-dimensional. Second, the implementation of these datatypes differ. The *vector* datatypes (`integerVector`, `floatVector`, `doubleVector`) are represented using a *list* simple datatype. Each entry in the vector is represented as one item in the list. The entries appear in the list in the same order as they occur in the vector. On the other hand, the matrix datatypes are represented using a complex datatype

whose content is a *list* simple datatype. The two datatypes overlap in that a matrix can represent a one-dimensional matrix, i.e. a vector.

To support variable-sized matrices the special attribute `mpeg7:dim` is used to specify the size of the dimensions of a matrix instance. The `mpeg7:dim` attribute is defined in DDL (see ISO/IEC 15938-2) as a list of positive integers. Each element in the list indicates the size of one dimension of the matrix. `mpeg7:dim` is a reserved attribute that shall be used to specify matrix dimensions and shall not be used for other purposes. For example, a value of "2 4" for `mpeg7:dim` indicates a 2-by-4 matrix having 2 rows and 4 columns. The `mpeg7:dim` attribute is only used for the matrix datatypes and not for the vector datatypes.

For describing a vector, the Vector datatypes (e.g. `integerVector`) or a 1-D Matrix (e.g. `IntegerMatrix`) can be used. For the matrix datatypes, `mpeg7:dim` is required. For the vector datatypes, `mpeg7:dim` is not allowed. Deciding which of these datatypes is appropriate depends on the needs of the application using the datatype.

3.3.4.2 vector datatypes

Information on extraction and use is not provided.

3.3.4.3 minusOneToOneVector datatype

3.3.4.4 Matrix datatypes

3.3.4.4.1 Matrix datatypes examples

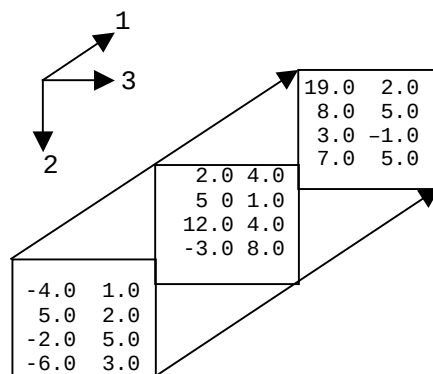
Consider the two-dimensional 4x4 matrix shown below:

$$\begin{bmatrix} 1.0 & 12.0 & 0.0 & 0.1 \\ 0.0 & 2.0 & 0.0 & 0.0 \\ 13.0 & 0.0 & -1.0 & 0.0 \\ 4.0 & 0.0 & 0.0 & 1.2 \end{bmatrix}$$

This matrix can be represented as the following instance of the `DoubleMatrix` datatype:

```
<DoubleMatrix mpeg7:dim="4 4">
  1.0 12.0  0.0 0.1
  0.0  2.0  0.0 0.0
 13.0  0.0 -1.0 0.0
  4.0  0.0  0.0 1.2
</DoubleMatrix>
```

Consider the following 3-dimensional matrix of size 3x4x2:



This 3-dimensional matrix can be represented as an instance of `IntegerMatrix` as follows:

```
<IntegerMatrix mpeg7:dim="3 4 2">
  <!-- The order of elements is
    (1,1,1), (1,1,2),
    (1,2,1), (1,2,2),
    (1,3,1), (1,3,2),
    (1,4,1), (1,4,2), . . .
```

```

      (2,1,1), (2,1,2), . . .
      (3,1,1), (3,1,2), . . .
      (3,4,1), (3,4,2), -->
-4   1   5   2  -2   5  -6   3
  2   4   5   1  12   4  -3   8
 19   2   8   5   3  -1   7   5
</IntegerMatrix>

```

3.3.4.5 Diagonal matrix datatypes

3.3.4.5.1 Diagonal matrix datatypes examples

Consider the diagonal matrix shown below:

$$\begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.2 \end{bmatrix}$$

Because this matrix is a diagonal matrix, it can be represented using the `DoubleDiagonalMatrix` datatype as follows:

```

<DoubleDiagonalMatrix mpeg7:dim="4">
  1.0 2.0 -1.0 1.2
</DoubleDiagonalMatrix>

```

3.3.4.6 MinusOneToOneMatrix datatype

Information on extraction and use is not provided.

3.3.5 Probability datatypes

3.3.5.1 probabilityVector datatype

Information on extraction and use is not provided.

3.3.5.2 ProbabilityMatrix datatype

3.3.5.2.1 ProbabilityMatrix datatype examples

Consider a conditional probability distribution involving three random variables *S*, *R*, and *W* each of which can have the value *true* or *false*. The conditional probability table for $P(W|S,R)$ is defined as follows:

Table 3- Example of conditional probabilities.

<i>R</i>	<i>S</i>	$P(W=false R,S)$	$P(W=true R,S)$
<i>false</i>	<i>false</i>	1.00	0.00
<i>false</i>	<i>true</i>	0.10	0.90
<i>true</i>	<i>false</i>	0.80	0.20
<i>true</i>	<i>true</i>	0.01	0.99

Then this conditional probability table is represented as a `ProbabilityMatrix` value as follows. In this example, *false* is represented as index 1 and *true* as index 2 and the random variables have been assigned to matrix dimensions as follows: *R* to dimension 1, *S* to dimension 2 and *W* to dimension 3. This results in the following value for `ProbabilityMatrix`:

```

<ProbabilityMatrix mpeg7:dim="2 2 2">
  1.0 0.0 <!-- P(W=false|R=false,S=false), P(W=true|R=false,S=false) -->
  0.1 0.9 <!-- P(W=false|R=false,S=true), P(W=true|R=false,S=true) -->
  0.8 0.2 <!-- P(W=false|R=true,S=false), P(W=true|R=true,S=false) -->
  0.01 0.99 <!-- P(W=false|R=true,S=true), P(W=true|R=true,S=true) -->

```

```
</ProbabilityMatrix>
```

Note that one could have used another assignment of variables to indexes rather than the one used in the example above.

3.3.6 String datatypes

3.3.6.1 mimeType datatype

Information on extraction and use is not provided.

3.3.6.2 countryCode datatype

3.3.6.2.1 countryCode datatype example

The following shows an example of using a countryCode. In this example, the country "Spain" is identified by its code "es".

```
<Country>es</Country>
```

3.3.6.3 regionCode datatype

3.3.6.3.1 regionCode datatype example

The following show examples of encoding places using region codes. Note that region codes are case insensitive.

```
<Place>
  <Name>Italian province of Milano</Name>
  <Region>IT-MI</Region>
</Place>

<Place>
  <Name>Danish county Roskilde</Name>
  <Region>dk-025</Region>
</Place>

<!-- This example purposely uses mixed case to show case-insensitivity of
country/region codes -->
<Place>
  <Name>Antananarivo province in Madagascar</Name>
  <Region>mG-t</Region>
</Place>
```

3.3.6.4 currencyCode datatype

3.3.6.4.1 currencyCode datatype example

The following shows several examples of different currency codes:

Currency Name	Currency Code
Hong Kong Dollar	HKD
Euro	EUR

3.3.6.5 characterSetCode datatype

3.3.6.5.1 characterSetCode datatype example

The following table shows several examples of character set codes:

Table 4 - Examples of character set codes.

Character Set	Code
Shift JIS encoding of Japanese character	Shift_JIS
RFC 2279 defined encoding of the ISO/IEC 10646 (Unicode) character set.	UTF-8

3.4 Linking, identification and localization tools

3.4.1 Introduction

This clause specifies a set of basic datatypes that are used for referencing within descriptions and linking of descriptions to multimedia content. While the DDL already includes a large library of built-in datatypes, the linking to multimedia data requires some additional datatypes, which are defined in this clause:

Table 5 - Overview of Linking Tools.

Tool	Functionality
References to Ds and DSs	Tool for representing references to parts of a description. The <code>ReferenceType</code> is defined as a referencing mechanism based on <code>anyURI</code> , <code>IDREF</code> or <code>xPathRefType</code> datatypes.
Unique Identifier	Tool for representing unique identifiers of content.
Time description tools	Tools for representing time specifications. Two formats are distinguished: the <code>TimeType</code> for time and date specifications according to the real world time and <code>MediaTimeType</code> for time and date specifications as they are used within media.
Media Locators	Tools for representing links to multimedia data.

3.4.2 References to Ds and DSs

3.4.2.1 Reference datatype

3.4.2.1.1 Reference datatype examples

The `Reference` datatype is used by description tools to make reference to other elements of a description. For example, the `Reference` datatype is used in relation graphs, and to describe summaries of multimedia content, and to relate descriptions of content structure and semantics (for examples, see clause 3.9 and clause 3.11). The examples below show the use of the `Reference` datatype for describing references using the three different mechanisms.

The first example shows the use of `idref` for referencing an element with the ID "id3" in the current description. A parser would be able to check if an attribute "id3" of type ID is present in the description.

```
<MyElement idref="id3">...</MyElement>
```

The second example shows the use of `xpath` for referencing the first `MyTargetElement` contained as a child after going up three times in the description tree along the parent axis from the current element which contains the specified `xpath` attribute.

```
<MyElement xpath="../../../../MyTargetElement[1]">...</MyElement>
```

The following example shows the use of `href` for referencing to an element with the ID "id1" in "anotherdescription.mp7".

```
<MyElement href="www.example.com/anotherdescription.mp7#id1"/>
```

3.4.2.2 XPath datatypes

Information on extraction and use is not provided.

3.4.3 Unique Identifier

3.4.3.1 UniqueID datatype

3.4.3.1.1 UniqueID datatype examples

The Unique Identifier datatype can be used when an identification of content is required, either the content or an instance of it. It can also be used as a unique reference to external entities, for example, to identify the rights associated with the content via a rights identifier belonging to an external IPMP system.

The following example show the use of the `UniqueID` datatype for identifying content. The first example shows the use of `UniqueID` datatype for identifying a book using the International Standard Book Number (ISBN).

```
<MyID type="ISBN">0-7803-5610-1</MyID>
```

The following examples shows the use of `UniqueID` datatype for describing an International Standard Work Code (ISWC).

```
<MyID organization="ISO" type="ISWC">T-034.524.680-1</MyID>
```

The following examples shows the use of `UniqueID` datatype for describing a URN as a unique identifier.

```
<MyID>urn:ietf:std:50</MyID>
```

Note that in this case the type and the organization are not specified in the attributes as they are already encoded in the identifier value itself. Furthermore, in this case the type of the identifier (URI) is the default value of "type" and does not need to be specified.

3.4.4 Time description tools

3.4.4.1 Time datatype

Information on extraction and use is not provided.

3.4.4.2 TimePoint datatype

Information on extraction and use is not provided.

3.4.4.3 Duration datatype

Information on extraction and use is not provided.

3.4.4.4 IncrDuration datatype

Information on extraction and use is not provided.

3.4.4.5 RelTimePoint datatype

Information on extraction and use is not provided.

3.4.4.6 RelIncrTimePoint datatype

Information on extraction and use is not provided.

3.4.4.7 TimeProperty attribute group

Information on extraction and use is not provided.

3.4.4.8 Time datatype examples

The following examples show the use of the `Time` datatypes. In a first example an event that started on the 3rd October 1987 at 14:13 in Germany and that has a duration of 10 days is described by:

```
<Time>
  <TimePoint>1987-10-03T14:13+01:00</TimePoint>
  <Duration>P10D</Duration>
</Time>
```

The following example describes the time at which a 3rd picture is taken given that one picture is taken each day:

```
<Time>
  <RelIncrTimePoint timeUnit="P1D"
timeBase="../../../../Time[1]">3</RelIncrTimePoint>
</Time>
```

The following example counts time units of one day and refers to the start time of the event as a timeBase (e.g. first time point specification in the description).

The period of five days after the initial event is described as follows:

```
<Time>
  <RelIncrTimePoint timeUnit="P1D"
timeBase="../../../../Time[1]">0</RelIncrTimePoint>
  <IncrDuration timeUnit="P1D">5</IncrDuration>
</Time>
```

The following example describes the time of an event occurring in England, two hours and 20 minutes after the initial one:

```
<Time>
  <RelTimePoint timeBase="../../../../Time[1]">PT2H20M-01:00Z</RelTimePoint>
</Time>
```

This specification is similar to the example using RelIncrTimePoint. But here the time offset is specified by using a number of hours and minutes instead of counting time units. Additionally it is specified that the time zone is different. Thus the local time is not +01:00 as it was the case for the initial event but +00:00 UTC.

Further examples of time expressions can be found in the clause on MediaTime. The Time datatype can be used whenever there is a need to describe a time segment whether it is a time point or a whole period.

3.4.4.9 MediaTime datatype examples

The MediaTime datatype can be used to describe a time segment whether it is a time point or a whole period using the time information of the AV content. For example the MediaTime datatype can be used within the MediaLocator datatype.

Consider a video consisting of the following segments:

- Segment1: 0-0.1(sec)
- Segment2: 0.1-1(sec)
- Segment3: 1-10 (sec)
- Segment4: 10-20 (sec)
- Segment5: 20-1000 (sec)

The video and these segments are described using the MediaTime datatype in different ways in these examples. Remind that this is done to explain the possibilities of the MediaTime datatype. In a description usually only one of these possibilities are used throughout the document.

Segment 1 is described by using the start time of the video as time base: the XPath expression is applied relative to the context node which is the element in which the mediaTimeBase attribute is contained. For instance "../../../../MediaLocator[1]" specifies the first MediaLocator contained in the parent node in which also the MediaTime element is contained. Accordingly the time line of the located media element is used to specify the video segment by the start time and the duration of the segment.

```
<!-- Specification of the video location -->
<MediaLocator>
  <MediaUri>http://www.mpeg7.org/the_video.mpg</MediaUri>
</MediaLocator>

<!-- MediaTime for Segment1(0-0.1sec) -->
<MediaTime>
```

```

    <MediaRelTimePoint
mediaTimeBase="../../MediaLocator[1]">PT0S</MediaRelTimePoint>
    <MediaDuration>PT1N10F</MediaDuration>
</MediaTime>

```

Segment 2 is also specified by the start time of the segment and the duration.

```

<!-- MediaTime for Segment2(0.1-1sec) -->
<MediaTime>
    <MediaRelTimePoint
        mediaTimeBase="../../MediaLocator[1]">PT1N10F</MediaRelTimePoint>
    <MediaDuration>PT9N10F</MediaDuration>
</MediaTime>

```

For segment 3 counting mediaTimeUnits are used for the specification of the start time and the duration. The mediaTimeUnit is specified as 1N which is 1/30 of a second according to 30F. But if needed a mediaTimeUnit for the exact sampling rate of NTSC of 29.97...Hz is also possible with: mediaTimeUnit="PT1001N30000F". For the representation of the exact sampling rate of 25Hz in the case of the PAL or SECAM video format the time unit shall be instantiated by mediaTimeUnit="PT1N25F".

```

<!-- MediaTime for Segment3(1-10sec) -->
<MediaTime>
    <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
        mediaTimeBase="../../MediaLocator[1]">30</MediaRelIncrTimePoint>
    <MediaIncrDuration mediaTimeUnit="PT1N30F">270</MediaIncrDuration>
</MediaTime>

```

The last segment again specifies the time using seconds and minutes.

```

<!-- MediaTime for Segment4(10-20sec) -->
<MediaTime>
    <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
        mediaTimeBase="../../MediaLocator[1]">300</MediaRelIncrTimePoint>
    <MediaIncrDuration mediaTimeUnit="PT1N30F">300</MediaIncrDuration>
</MediaTime>

```

```

<!-- MediaTime for Segment5(20-1000sec) -->
<MediaTime>
    <MediaRelTimePoint
mediaTimeBase="../../MediaLocator[1]">PT20S</MediaRelTimePoint>
    <MediaDuration>PT16M20S</MediaDuration>
</MediaTime>

```

The duration may be described by counting mediaTimeUnits (in this case seconds) using MediaIncrDuration as follows:

```

<MediaIncrDuration mediaTimeUnit="PT1S">980</MediaIncrDuration>

```

For another example, consider a description of a video segment with its root segment specified with:

```

<MediaTime>
    <MediaTimePoint>1989-10-03T14:13:02:0F30</MediaTimePoint>
    <MediaIncrDuration mediaTimeUnit="PT1N30F">1340</MediaIncrDuration>
</MediaTime>

```

Since the video itself is assumed to have a frame rate of 30 frames per second, fractions of one second is set to 30F and the mediaTimeUnit is set to 1N. Thus, by specifying a segment of 1340 frames, the duration of the segment is also described. If needed the date can also be specified within the MediaTime. To specify a single frame in a video sequence, the frame numbers or the timestamps can be used in the following way:

```
<MediaTime>
  <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
    mediaTimeBase="../../MediaTime[1]">561</MediaRelIncrTimePoint>
</MediaTime>
```

In this last case, the 561st frame is referenced using the very beginning of the described segment as a start point (it is assumed that this is the first occurrence of a MediaTime Element). The video itself is displayed at a frame rate of 30 frames per second. In the example each frame is counted (1N) thus the timestamp is 18s:21.

Alternatively, timestamps can be used directly to relate elapsed time to the counted frames as follows:

```
<MediaTime>
  <MediaRelTimePoint mediaTimeBase="../../MediaTime[1]">PT18S21N30F
</MediaRelTimePoint>
</MediaTime>
```

Consider the case in which the video is divided into subsegments (e.g. a shot from 500 to 600) as follows:

```
<MediaTime>
  <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
    mediaTimeBase="../../MediaTime[1]">500</MediaRelIncrTimePoint>
  <MediaIncrDuration mediaTimeUnit="PT1N30F">100</MediaIncrDuration>
</MediaTime>
```

A frame can be addressed within this segment as follows:

```
<MediaTime>
  <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
    mediaTimeBase="../../../MediaTime">61</MediaRelIncrTimePoint>
</MediaTime>
```

In this case the start time of the shot (e.g. the parent node of the node which contains the present MediaTime datatype) is referenced as mediaTimeBase.

To avoid having to repeatedly describe the mediaTimeUnit and the mediaTimeBase, the mediaTimeUnit and mediaTimeBase attribute can be used in VideoSegment as follows:

```
<VideoSegment mediaTimeUnit="PT1N30F" mediaTimeBase="MediaLocator">
  <!-- more elements here -->
</VideoSegment>
```

If the specified mediaTimeUnit and mediaTimeBase attributes are instantiated in a video segment that is an ancestor of this instance, then the frame can be addressed as follows:

```
<VideoSegment mediaTimeUnit="PT1N30F" mediaTimeBase="MediaLocator">
  <TemporalDecomposition>
    <VideoSegment id="frame561">
      <MediaTime>
        <MediaRelIncrTimePoint>61</MediaRelIncrTimePoint>
      </MediaTime>
    </VideoSegment>
  </TemporalDecomposition>
</VideoSegment>
```

3.4.5 Media Locators

3.4.5.1 MediaLocator datatype

3.4.5.2 InlineMedia datatype

3.4.5.2.1 InlineMedia datatype examples

The InlineMedia datatype enables for instance an embedding of small audio clips and image thumbnails in the description, without having to refer to external data. An example is as follows.

```
<MyInlineMedia type="image/jpeg">
  <MediaData16>98A34F10C5094538AB9387362522DA3</MediaData16>
</MyInlineMedia>
```

3.4.5.3 TemporalSegmentLocator datatype

Information on extraction and use is not provided.

3.4.5.4 ImageLocator datatype

Information on extraction and use is not provided.

3.4.5.5 MediaLocator datatype examples

In the following example the location of a video segment is specified by the URI of a video file and the relative start time with respect to the beginning of the file and the duration of the segment.

```
<MyTemporalSegmentLocator>
  <MediaUri>http://www.mpeg7.org/demo.mpg</MediaUri>
  <MediaTime>
    <MediaRelTimePoint
mediaTimeBase="../../MediaUri">PT3S</MediaRelTimePoint>
    <MediaDuration>PT10S</MediaDuration>
  </MediaTime>
</MyTemporalSegmentLocator>
```

In the second example, the MediaLocator refers to a particular stream within an MPEG-4 file.

```
<MyMediaLocator>
  <MediaUri>http://www.mpeg7.org/demo.mp4</MediaUri>
  <StreamID>1</StreamID>
</MyMediaLocator>
```

In the third example, the media data is simply located using a byte-offset from the start of a known file.

```
<MyImageLocator>
  <MediaUri>http://www.mpeg7.org/demo.ppm</MediaUri>
  <BytePosition offset="1024"/>
</MyImageLocator>
```

3.5 Basic description tools

3.5.1 Introduction

This clause defines the basic tools, which are used as components for building other description tools. The basic tools are used in the Multimedia Description Schemes part of the standard (ISO/IEC 15938-5) and also in audio part (see ISO/IEC 15938-4) and visual (see ISO/IEC 15938-3) part. The tools defined in this clause are as follows.

Table 6 - Overview of Basic Description Tools.

<i>Tool</i>	<i>Functionality</i>
Language Identification	Tools for identifying the language of a textual description or of the multimedia content itself. This standard uses the XML defined <code>xml:lang</code> attribute to identify the language used to write a textual description.
Text Annotation	Tools for representing unstructured and structured textual annotations. Unstructured annotations (i.e. with free text) are represented using the <code>FreeTextAnnotation</code> datatype. Annotations that are structured in terms of answering the questions "Who? What? Where? How? Why?" are represented using the <code>StructuredAnnotation</code> datatype. Annotations structured as a set of keywords are representations using the <code>KeywordAnnotation</code>

	datatype. Finally, annotations structured by syntactic dependency relations—for example, the relation between a verb phrase and its subject—are represented using the <code>DependencyStructure</code> datatype.
Classification Schemes and Terms	Tools for classifying using language-independent terms and for defining sets of terms as classification schemes. The <code>ClassificationScheme</code> DS describes a vocabulary for classifying a subject area as a set of terms organized into a hierarchy. A term defined in a classification scheme is used in a description with the <code>TermUse</code> or <code>ControlledTermUse</code> datatypes. Graphical classification schemes are schemes for classifying where the terms in the scheme are themselves graphs. Such schemes can be used as structural templates, for validating of graph-based descriptions, and for graph productions.
Agents	Tools for describing "agents", including persons, groups of persons, and organizations. The <code>Person</code> DS represents a person, the <code>PersonGroup</code> DS a group of persons, and the <code>Organization</code> DS an organization of people.
Places	Tools for describing geographical locations. The <code>Place</code> DS is used to describe real and fictional places.
Graphs and Relations	Tools for representing relations and graph structures. The <code>Relation</code> DS is a tool for representing named relations between instances of description tools. The <code>Graph</code> DS organizes relations into a graph structure.
Ordering	The <code>OrderingKey</code> DS describes hints for ordering descriptions
Affective Description	The <code>Affective</code> DS describes an audience's affective response to multimedia content.
Phonetic Description	The <code>PhoneticTranscriptionLexicon</code> DS describes the pronunciations of a set of words.

3.5.2 Language identification

3.5.2.1 xml:lang attribute

3.5.2.1.1 xml:lang attribute examples

Consider the following example:

```
<TextAnnotation xml:lang="en">
  <KeywordAnnotation>
    <Keyword xml:lang="fr"> chien </Keyword>
    <Keyword> dog </Keyword>
  </KeywordAnnotation>
</TextAnnotation>
```

In this example, the outer level `TextAnnotation` specifies that the language of the description is English ("en"). This automatically applies to all elements contained and thus the value of the `xml:lang` for the contained `Keyword` "dog", defaults to be English. However, because the keyword "chien" specifies that the language is French ("fr") using `xml:lang`, this value overrides the language specified in the containing `TextAnnotation`.

3.5.2.2 language datatype

3.5.2.2.1 language datatype examples

The `xsd:language` datatype is used for identifying the language properties of the multimedia content. For example, to indicate the language used in the audio, the written language of subtitles, or a user's preferred language for viewing content.

The example below states that the preferred language is English.

```
<PreferredLanguage>en</PreferredLanguage>
```

3.5.3 Textual annotation

3.5.3.1 Textual datatypes

3.5.3.1.1 Textual datatypes example

For examples of the use of these datatypes, see the subclause specifying the TextAnnotation datatype.

3.5.3.2 TextAnnotation datatype

3.5.3.2.1 TextAnnotation datatype examples

The following example shows an annotation of a video depicting the scene "Tanaka throws a small ball to Yamada. Yamada catches the ball with his left hand."

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <MediaLocator>
          <MediaUri>video.mpg</MediaUri>
        </MediaLocator>
        <TextAnnotation>
          <FreeTextAnnotation xml:lang="en">
            Tanaka throws a small ball to Yamada.
            Yamada catches the ball with his left hand.
          </FreeTextAnnotation>
          <StructuredAnnotation>
            <Who href="urn:people:TANAKA100">
              <Name xml:lang="en">Tanaka</Name>
            </Who>
            <Who>
              <Name xml:lang="en"> Yamada </Name>
            </Who>
            <WhatObject>
              <Name xml:lang="en"> A small ball </Name>
            </WhatObject>
            <WhatAction>
              <Name xml:lang="en"> Tanaka throws a small ball to
Yamada.
              </Name>
            </WhatAction>
            <WhatAction>
              <Name xml:lang="en"> Yamada catches the ball with his
left
              hand. </Name>
            </WhatAction>
          </StructuredAnnotation>
        </TextAnnotation>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

In this example, there are two different forms of the annotation for the same event:

- **Free text annotation.** This is simply an English description of what happened in the scene without any structuring.
- **Structured text annotation.** In this case the annotation is more structured; "Yamada" is identified as the "Who" and the ball he is throwing as the "What" in the annotation. Also, notice

that Yamada is identified using a term "urn:people:TANAKA100" from a classification scheme for identifying people (identified as "urn:people"). .

3.5.3.3 StructuredAnnotation datatype

Information on extraction and use is not provided.

3.5.3.4 KeywordAnnotation datatype

3.5.3.4.1 KeywordAnnotation datatype examples

The following example shows an annotation describing a video about the U.S. presidential election using keywords:

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <TextAnnotation>
          <KeywordAnnotation xml:lang="en">
            <Keyword>election</Keyword>
            <Keyword>United States of America</Keyword>
            <Keyword>President</Keyword>
          </KeywordAnnotation>
        </TextAnnotation>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

3.5.3.5 DependencyStructure datatype

3.5.3.5.1 DependencyStructure datatype examples

The following rules specify the dependency relations in English:

Governor	Dependent	Operator/FunctionWord (of DependencyStructure datatype)	Example (G: Governor, D: Dependent)
Verb	Noun	subject	<u>John</u> (D) <u>drank</u> (G)
Verb	Noun	object	<u>drink</u> (G) <u>water</u> (D)
Verb	Noun	indirectObject	<u>tell</u> (G) <u>you</u> (D) the fact
Verb	Preposition	functionWord=(Preposition)	<u>respect</u> (G) <u>for</u> (D) John <u>sit</u> (G) <u>on</u> (D) a chair
Verb	Noun Adjective Preposition	predicate	<u>make</u> (G) it <u>possible</u> (D) <u>are</u> (G) <u>friends</u> (D) <u>is</u> (G) <u>without</u> (D) a hat John <u>returned</u> (G) <u>rich</u> (D)
Verb	Adverb		<u>walk</u> (G) <u>fast</u> (D)
Verb	Conjunction	functionWord=(Conjunction)	<u>started</u> (G) <u>before</u> (D) I finished breakfast
Noun	Adjective		<u>good</u> (D) <u>friends</u> (G)
Noun	Genitive		<u>John's</u> (D) <u>friends</u> (G)
Noun	Determiner		<u>the</u> (D) <u>cake</u> (G)
Noun	Noun		<u>London</u> (D) <u>hospital</u> (G)

Governor	Dependent	Operator/FunctionWord (of DependencyStructure datatype)	Example (G: Governor, D: Dependent)
Noun	Preposition	functionWord=(Preposition)	<u>years</u> (G) <u>of</u> (D) war
Noun	Verb	functionWord=(Relative-pronoun)	<u>people</u> (G) who <u>think</u> (D) that
Adjective	Adverb		<u>very</u> (D) <u>good</u> (G)
Preposition	Noun		<u>in</u> (G) <u>water</u> (D)
Conjunction	Verb		<u>before</u> (G) I <u>finished</u> (D) breakfast
Auxiliary verb	Verb		<u>can</u> (G) <u>read</u> (D) a book
"to"	Verb		<u>to</u> (G) <u>go</u> (D) to bed

Rules for other languages can be specified in a similar fashion but are not specified by this standard.

Consider the dependency structure of the following sentence: "John talks to Mary whom he loves." According to the rule above, the following tree structure (syntactic tree) represents the dependency structure of the sentence.

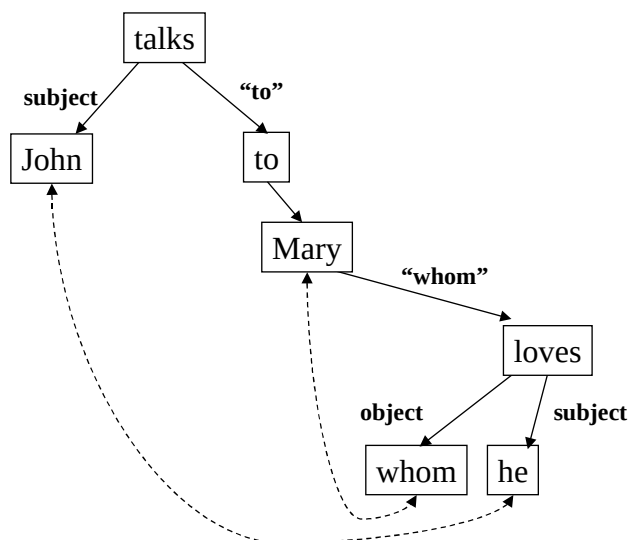


Figure 1 - Example of dependency structure of the sentence: "John talks to Mary whom he loves."

The solid arrows in Figure 1 show dependency relations and the dashed arrows indicate coreference relations. The labels attached to the arrows indicate the type of dependency relation. Specifically, the verb "talks" governs "John" and "to (Mary)", while "Mary" governs a verb "loves" in a relative clause. In the relative clause, the verb "loves" governs "whom", which refers "Mary" in a main clause, and "he", which refers to "John".

The following shows the example represented using the DependencyStructure datatype:

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <TextAnnotation>
          <DependencyStructure xml:lang="en">
            <Sentence>
              <Phrase operator="subject">
                <Head type="noun" id="JOHN">John</Head>
              </Phrase>
              <Head type="verb" baseForm="talk">talks</Head>
            </Sentence>
          </DependencyStructure>
        </TextAnnotation>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
  
```

```

        <Phrase functionWord="to">
          <Head type="preposition">to</Head>
          <Phrase>
            <Head type="noun" id="MARY">Mary</Head>
            <Phrase functionWord="whom">
              <Phrase operator="object">
                <Head type="pronoun"
equal="MARY">whom</Head>
              </Phrase>
            <Phrase operator="subject">
              <Head type="pronoun" equal="JOHN">he</Head>
            </Phrase>
            <Head type="verb" baseForm="love">loves</Head>
          </Phrase>
        </Phrase>
      </Sentence>
    </DependencyStructure>
  </TextAnnotation>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

Notice that the preposition "to" is represented as a functionWord attribute attached to the Phrase element representing "to Mary who he loves".

3.5.3.5.2 DependencyStructure datatype extraction

Using a natural language parser (NL parser), it is possible to generate the dependency structure for manually generated free text annotations. For example, one can automatically extract the following information:

Dependency structures

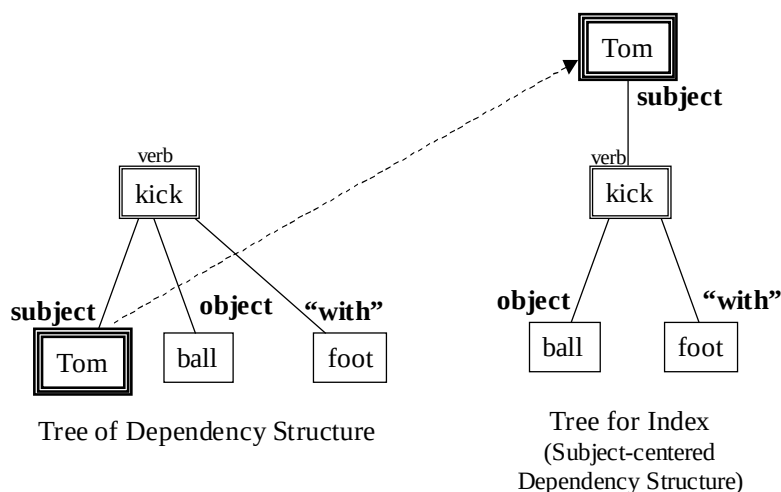
Syntactic role of phrases in the annotation-sentences (relationship with governor is represented by the particle attribute and the operator attribute).

In general, natural language syntactic analysis involves some ambiguity, i.e., a unique solution cannot always be given. The constrained annotations generated by professionals have a relatively simple structure and high accuracy can generally be obtained. On the other hand, amateur annotations, which tend to include many modifiers that make sentences long, results in relatively low analysis accuracy.

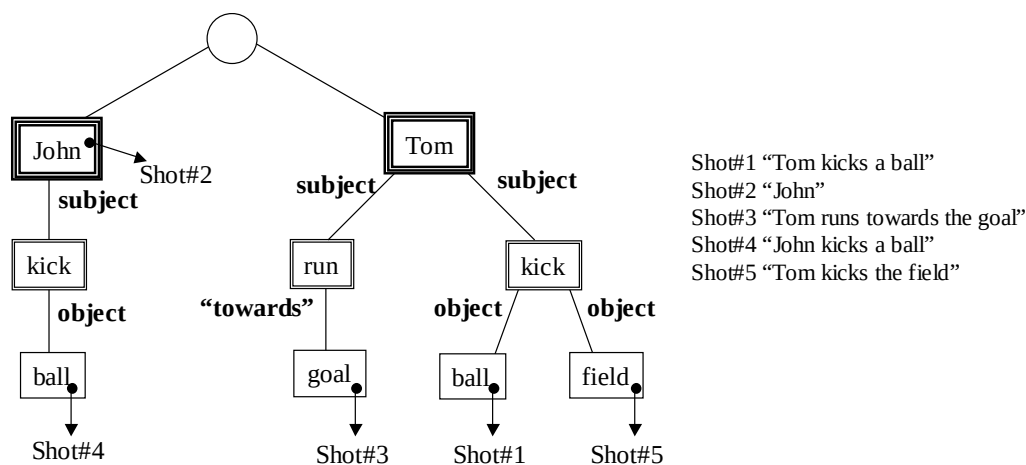
Human correction for erroneous results of the syntactic analysis is usually needed. Most of the errors come from the syntactic ambiguity about which phrase modifies which term. Correcting this type of error needs semantic analysis of the sentences. But the error can be fixed by simply changing a governor term (i.e., head) of a modifier phrase (i.e., dependent), which is not very complicated for humans. The syntactic analysis by the NL parser is helpful for instantiation of the Dependency Structure not only from the professional style annotations but also from the amateur style ones that include syntactic ambiguity.

One can retrieve audio or video segments using natural language as an interface by describing video segment with the DependencyStructure datatype. To do this a video segment index is prepared according to the following procedure.

For each segment, the syntactic tree representing an instance of the Dependency Structure is converted to a tree structure for index use. In the tree structure, a noun that is a head of a "sentence" or a head of a phrase that depends on a predicate of a "sentence" as a subject is taken to be the root node. If no such term exists, a dummy node is added.



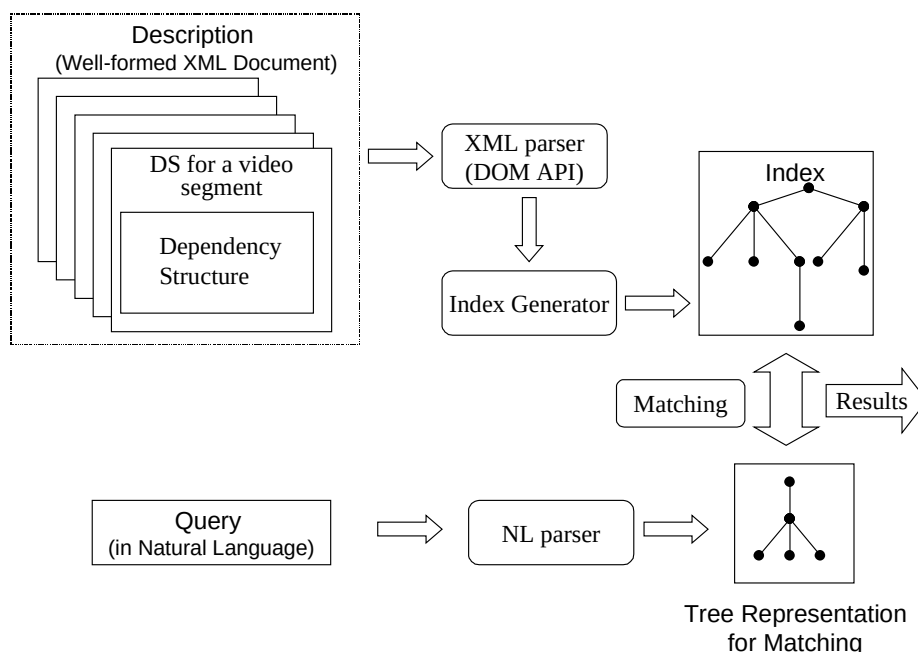
All tree structures for the segment index included in the database are merged to form a database index tree.



Each node in the index tree stores a pointer to the segment in question. Therefore, to perform a search using this index tree, a natural language search query is first converted to a tree structure. This tree is then matched with the index trees to find the desired segment.

Matching may be achieved by partial tree matching and, in the case of inter-node matching, similarity-based retrieval can be performed by using a thesaurus or the like. The reason for configuring an index tree centered about the subject/theme is that candidate can be narrowed down from the start through the use of subjects/themes, which provide the most information in keyword-centered annotations.

An example system is configured as follows.



Traditionally, keywords are used when retrieving based on language. However, using the Dependency Structure has the following advantages.

It supports retrieval that specify inter-element relationships such as "A gives B to C" which cannot be expressed solely by Boolean operators (AND/OR) as employed in ordinary keyword retrieval.

It supports flexible retrieval by allowing the use of "wild cards" such as in the expression "C eats something."

It requires no special query language (retrieval can be performed with natural language sentences).

3.5.4 Classification Schemes and Terms

3.5.4.1 ClassificationScheme DS

3.5.4.1.1 ClassificationScheme format and importing

Information on extraction and use is not provided.

3.5.4.1.2 ClassificationScheme DS examples

The following subclauses show several different examples of defining classification schemes.

Classification Scheme Example One: Audio Domain

The following example shows a classification scheme for audio domain. Note that while this example is based on the the AudioDomainCS classification specified in Annex A, it has been modified and simplified here for explanatory purposes. The instance of the ClassificationScheme DS defining the scheme looks as follows:

```
<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationScheme uri="urn:mpeg:mpeg7:cs:AudioDomainCS"

    domain="//MediaInformation/MediaProfile/MediaIdentification/AudioDomain">
      <Header xsi:type="DescriptionMetadataType">
        <Version>1.0</Version>
        <Rights>
          <RightsID>urn:rights:x0d01d</RightsID>
        </Rights>
      </Header>
      <Term termID="1">
        <Name xml:lang="en">Source</Name>
        <Definition xml:lang="en">Type of audio source</Definition>
        <Term termID="1.1">
          <Name xml:lang="en">Synthetic</Name>
        </Term>
        <Term termID="1.2">
```

```

        <Name xml:lang="en">Natural</Name>
      </Term>
    </Term>
    <Term termID="2">
      <Name xml:lang="en">Acquisition</Name>
      <Definition xml:lang="en">Type of Content</Definition>
      <Term termID="2.1">
        <Name xml:lang="en">Music</Name>
      </Term>
      <Term termID="2.2">
        <Name xml:lang="en">Speech</Name>
        <Name xml:lang="jp">Onsei</Name>
      </Term>
      <Term termID="2.3">
        <Name xml:lang="en">Mixed</Name>
      </Term>
      <Term termID="2.4">
        <Name xml:lang="en">Multi-track</Name>
      </Term>
    </Term>
    <Term termID="3">
      <Name xml:lang="en">Use</Name>
      <Definition xml:lang="en">Original usage domain</Definition>
      <Term termID="3.1">
        <Name xml:lang="en">Scientific</Name>
      </Term>
      <Term termID="3.2">
        <Name xml:lang="en">Medical</Name>
      </Term>
      <Term termID="3.3">
        <Name xml:lang="en">Security</Name>
      </Term>
      <Term termID="3.4">
        <Name xml:lang="en">Broadcast</Name>
      </Term>
      <Term termID="3.4">
        <Name xml:lang="en">CD-A distribution</Name>
      </Term>
    </Term>
  </ClassificationScheme>
</Description>
</Mpeg7>

```

The first part of the ClassificationScheme instance contains "header" information including:

- **URI.** The unique identifier for this ClassificationScheme is a URN: "urn:mpeg:mpeg7:cs:AudioDomainCS". The first part of the URI "urn:" indicates that this URI is a URN. In fact, all classification schemes defined in this document use URNs as their unique identifier. The second part of the URN, "mpeg", is the namespace identifier. In this case the namespace identifier ("mpeg") is one that is registered as belonging to MPEG. Following the "mpeg" namespace identifier is the namespace specific string "mpeg7", which identifies the classification schemes as being defined in this standard and a "cs" string, which indicates that name, is for a classification scheme. Finally, the "AudioDomainCS" is the name assigned to the classification scheme within the "mpeg" URN namespace.
- **Domain.** The suggested domain for this ClassificationScheme is "MediaInformation/MediaProfile/MediaIdentification/AudioDomain". The domain is expressed as an XPath value. In this case, the XPath matches the AudioDomain element defined inside of the MediaIdentification DS (contained within the MediaProfile DS in the MediaInformation DS).
- **Description Metadata.** This element contains metadata about the ClassificationScheme itself. In the example, the description metadata indicates that this is version "1.0" of the

classification scheme and that the intellectual property rights for the classification scheme are identified by "urn:rights:x0d01d".

The next part of this ClassificationScheme contains a list of term definitions. This list includes:

- **Term Identifiers.** In this classification scheme, the term identifiers are numerals that encode the term hierarchy. However, this use is simply a convention adopted in this classification scheme. In fact, term identifiers are simply opaque values whose only requirement is that they be unique within the classification scheme. For example, instead of the term identifier "1" for the term "Source", the term identifier might equally well have been "0d3d" or "source".
- **Term Names.** Each term in this classification scheme has an English name (the language is indicated by the `xml:lang` attribute having a value of "en"). The term names are human readable labels for the terms. Note that terms may have names in multiple languages. For example, term "2.2" has a name in both English ("Speech") and Japanese ("Onsei").
- **Term Definitions.** Only terms "1" and "2" have prose definitions in this classification scheme. The other terms do not have written definitions, which is allowed because the `Definition` element in a term is optional.
- **Term Relations.** This classification scheme contains a hierarchy of terms. In other words, it defines broader term/narrower term relations between pairs of terms. For example, term "1" is a broader term than term "1.1" and term "1.1" is a narrower term than term "1" – these are inverse relations. Note that the definition of term "1.1" is syntactically contained within the definition of term "1". This is how term relations are written in a ClassificationScheme. In fact, other kinds of relations can be specified by using the "relation" attribute to relate the parent and contained terms. Because the default value of a term relation is "NT", indicating that the contained term is narrower than the containing term.

Classification Scheme Example Two: Genre

The following figure shows an example of a simple classification for genre identified by the URI "urn:example:Genre". In the figure below, round boxes represent classification schemes and square boxes represent terms.

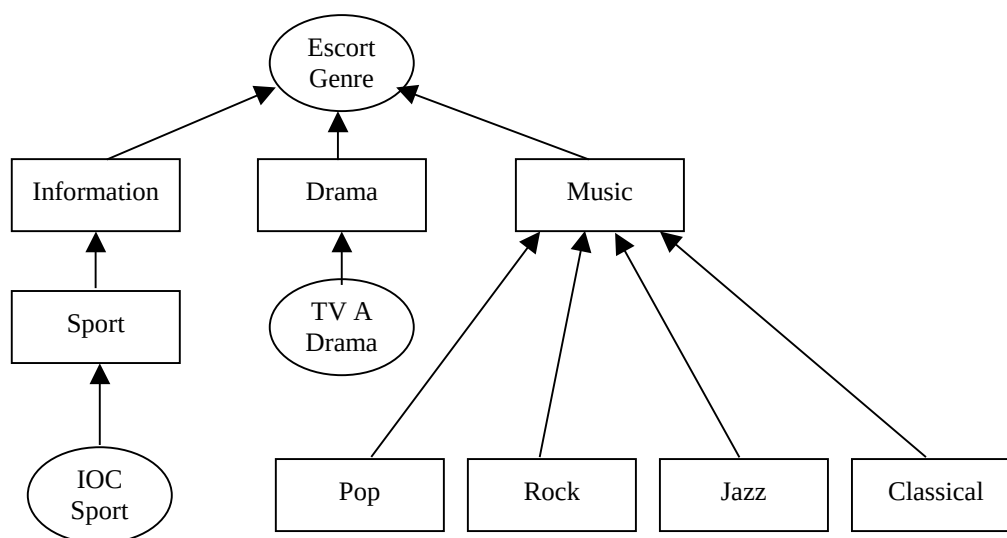


Figure 2 - Illustration of the "Escort2_4: Genre" Genre Classification Scheme.

In this example, there are three terms at the highest level: information, drama, and music. Below the category "information", there is a more detailed term: "Sport". Rather than defining a complete classification for sports, this example shows how an existing classification scheme can be imported. The existing "IOC Sport" classification scheme is added below the "Sport" item. Similarly, the drama category imports the "TVE Drama" classification scheme. For music, the sub-terms (narrower in meaning than their containing term) are "Pop", "Rock", "Jazz", and "Classical".

```

<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationScheme uri="urn:example:Genre"
      domain="//CreationInformation/Classification/Genre">
      <Import href="http://www.tvid.org/TVE/Drama/" />
    </ClassificationScheme>
  </Description>

```

```

    <Import href="http://www.ioc.org/IOC/Sports/" />
    <Term termID="information">
      <Name xml:lang="en">Information</Name>
      <Definition xml:lang="en">Generic news</Definition>
      <!-- This narrower term comes from the imported IOC Sports scheme
-->
      <Term termID="sports" />
    </Term>
    <Term termID="drama">
      <!-- This narrower term comes from the imported drama scheme-->
    </Term>
    <Term termID="music">
      <Term termID="rock">
        <Name xml:lang="en">Rock</Name>
      </Term>
      <Term termID="pop">
        <Name xml:lang="en">Pop</Name>
      </Term>
      <Term termID="jazz">
        <Name xml:lang="en">Jazz</Name>
      </Term>
      <Term termID="classical">
        <Name xml:lang="en">Classical</Name>
      </Term>
    </Term>
  </ClassificationScheme>
</Description>
</Mpeg7>

```

3.5.4.2 TermDefinition DS

Information on extraction and use is not provided.

3.5.4.3 TermUse datatype

3.5.4.3.1 TermUse datatype examples

3.5.4.3.1.1 Term reference use

The following example shows some examples of *term reference* using several different means to refer to terms:

```

<!-- Referring to a term using a URN -->
<TermUse href="urn:mpeg:mpeg7:cs:GenreCS:alternativeJazz"/>

<!-- Referring to a term using a HTTP URI -->
<TermUse href="http://www.mpeg.org/GenreCS.xml#alternativeJazz"/>

<!-- Referring to a term using a classification scheme alias -->
<!-- Define schema aliases -->
<ClassificationSchemeAlias alias="s1" href="urn:mpeg:mpeg7:cs:GenreCS"/>
<ClassificationSchemeAlias alias="s2" href="http://www.mpeg.org/GenreCS.xml"/>

<!-- Refer to the term using the two schema aliases -->
<TermUse href="s1:alternativeJazz"/>
<TermUse href="s2:alternativeJazz"/>

```

The reference to the term is included in the href attribute. In the first example, this is the URN "urn:mpeg:mpeg7:cs:GenreCS:alternativeJazz". Using URN's to reference terms defined in classification schemes is preferred in general and shall be used for referring to terms defined in this document. However, terms defined in existing classification managed by other organizations can be referred to using any valid URI. For example, the reference "http://www.mpeg.org/GenreCS.xml#alternativeJazz" uses a HTTP format for referring to a term. The final two example of TermUse show the use of classification schemes aliases, which provide an

abbreviated syntax for referring to terms. In this case, aliases "s1" is assigned to the classification scheme identified by "urn:mpeg:mpeg7:cs:GenreCS" and the alias "s2" is assigned to the classification at "http://www.mpeg.org/GenreCS.xml". For further details on term references, see the specification of the termReference datatype.

3.5.4.3.1.2 Inline term use

This next example shows an *inline* term use.

```
<TermUse href="urn:AISF:Sports:1.4.9">
  <Name xml:lang="en">Basketball</Name>
  <Definition xml:lang="en">Basket Description</Definition>
  <Term relation="BT" termID="1.4">
    <Name xml:lang="en">Sport</Name>
    <Definition xml:lang="en">Sports news</Definition>
    <Term relation="BT" termID="1">
      <Name xml:lang="en">Information</Name>
      <Definition xml:lang="en">Generic news</Definition>
    </Term>
  </Term>
</TermUse>
```

In this example, the term "1.4.9" (i.e. basketball) is taken from the "urn:GAISF:Sports" classification scheme. In addition to the term identifier in the href attribute, the TermUse also includes inline information about the term: its name, definition, and some related terms. In this example, the term "1.4.9" is related to the broader term "1.4", which is in turn related to the broader term "1". In other words, this indicates that the hierarchy of terms for the genre is: 1.4.9 (Basketball) --> Sports (1.4) --> 1 (Information) .

3.5.4.3.1.3 Free term use

The last example shows the use of TermUse datatype as a *free term use*.

```
<TermUse>
  <Name>My genre</Name>
</TermUse>
```

What distinguishes a free term from a reference TermUse or an inline TermUse is that there is no reference to a term in a classification scheme. Instead of referencing the term, a free term use includes an arbitrary textual name or definition. This use is supported to allow the option of either writing free text or using a term inside of a description. It provides the greatest freedom of expression as no restrictions are imposed. On the other hand, this means that an application may find it difficult to interpret the content of a free TermUse.

3.5.4.4 ControlledTermUse datatype

3.5.4.4.1 ControlledTermUse datatype examples

The following example shows how a ControlledTermUse can refer to a term in an existing classification scheme, which has not been defined as an instance of the ClassificationScheme DS. The first example is taken from The Art and Architecture Thesaurus (AAT) compiled by the Getty Research Institutes. The AAT is a thesaurus widely used for cataloguing in the museum community. This case shows how the term for "High Baroque" in the AAT can be referenced as a controlled term:

```
<ControlledTermUse href="http://www.getty.edu/aat/21150">
  <Name xml:lang="en">High Baroque</Name>
  <Definition>The fully Baroque style that emerged in ...</Definition>
</ControlledTermUse>
```

Note several things about this example:

- **Term Reference.** In this example, the resource for the term is identified by a URI. In the example, the URI identifies the source (the Getty Research Institute) and the term's identifier ("21150") within the AAT. In the AAT, term identifiers are opaque numeric values.
- **Name and Definition.** Names and definitions can be given for terms in a classification schemes by including them inline inside of the ControlledTermUse. In this case the name indicates that

the term means "High Baroque" in English and gives a definition of that term in English (abbreviated in the example).

The second example of referencing a term in an existing classification scheme uses a term in the Dewey Decimal classification scheme (used in many libraries for cataloguing books).

```
<ControlledTermUse href="urn:dewey:630">
  <Name xml:lang="en">Agriculture</Name>
</ControlledTermUse>
```

The term is referenced using a URN (made up for the purpose of this example), which indicates that the term comes from the "dewey" URN identifier space. The "630" following the colon further qualifies the in the "dewey" URN space. In this case it indicates that it refers to the "630"; that is, pertaining to agriculture.

3.5.4.5 termReference datatype

3.5.4.5.1 Format of URN for terms

The following defines the normative form of URN for referring to terms defined in classification schemes specified in this or other parts of the ISO/IEC 15938 standard.

If the termReferenceType is a URI, and the term being referenced is defined in ISO/IEC 15938, whether normative or informative, then the URI shall be a URN of the following form:

```
urn:mpeg:mpeg7:cs:SchemeName:termID"
```

The "urn:mpeg:mpeg7:cs:SchemeName" is the URN identifying the classification scheme for this term, as specified in 3.5.4. The *termID* part of the URN shall match the *termID* attribute value specified for the term in the term's definition. Note however that URN's defined in the MPEG URN namespace are case insensitive and therefore the use of upper and lowercase letters are not distinguished in both *termID* and *schemeName*

For classification schemes that are not defined in ISO/IEC 15938, the form of the URI used to reference terms is not specified by this standard. However, using a URN for term reference URIs is strongly recommended.

3.5.4.6 ClassificationSchemeAlias datatype

Information on extraction and use is not provided.

3.5.4.7 GraphicalClassificationScheme DS

Information on extraction and use is not provided.

3.5.4.8 GraphicalTermDefinition DS

Information on extraction and use is not provided.

3.5.4.9 GraphicalRuleDefinition DS

Information on extraction and use is not provided.

3.5.4.10 PullbackDefinition DS

Information on extraction and use is not provided.

3.5.4.11 DoublePullbackDefinition DS

Information on extraction and use is not provided.

3.5.4.12 PushoutDefinition DS

Information on extraction and use is not provided.

3.5.4.13 DoublePushoutDefinition DS

Information on extraction and use is not provided.

3.5.4.14 GraphicalTermDefinition DS examples

The complete example description follows:

```
<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationSchemeBase xsi:type="GraphicalClassificationSchemeType"
      uri="urn:example:cs:Graph1">
```