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

**Part 5:
Multimedia description schemes**

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

Partie 5: Schémas de description multimédia



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.

STANDARDSISO.COM : Click to view the full PDF of ISO/IEC 15938-5:2003

© ISO/IEC 2003

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

Page

Foreword.....	v
Introduction	vi
1 Scope	1
1.1 Organization of the document.....	1
1.2 Overview of Multimedia Description Schemes.....	1
2 Normative references	3
3 Terms and definitions	4
3.1 Conventions	4
3.2 Wrapper of the schema.....	6
3.3 Terminology	6
3.4 Symbols and abbreviated terms	11
4 Schema tools	12
4.1 Introduction.....	12
4.2 Base types.....	12
4.3 Root element.....	16
4.4 Top-level types.....	19
4.5 Description metadata tools.....	42
5 Basic datatypes	48
5.1 Introduction.....	48
5.2 Integer datatypes	48
5.3 Real datatypes	54
5.4 Vectors and matrices	56
5.5 Probability datatypes	64
5.6 String datatypes.....	67
6 Linking, identification and localization tools.....	71
6.1 Introduction.....	71
6.2 References to Ds and DSS.....	72
6.3 Unique Identifier	75
6.4 Time description tools	77
6.5 Media Locators	95
7 Basic description tools	101
7.1 Introduction.....	101
7.2 Language identification	102
7.3 Textual annotation.....	103
7.4 Classification Schemes and Terms	119
7.5 Description of agents.....	158
7.6 Description of places	171
7.7 Graphs and relations.....	175
7.8 Ordering Tools.....	187
7.9 Affective description	190
7.10 Phonetic description	194
8 Media description tools.....	196
8.1 Introduction.....	196
8.2 Media information tools	198
9 Creation and production description tools.....	222
9.1 Introduction.....	222
9.2 Creation information tools.....	222
10 Usage description tools	237
10.1 Introduction.....	237

10.2	Usage information tools.....	237
11	Structure description tools.....	245
11.1	Introduction	245
11.2	Base segment description tools	252
11.3	Segment attribute description tools	262
11.4	Visual segment description tools	278
11.5	Audio segment description tools.....	308
11.6	Audio-visual segment description tools	312
11.7	Multimedia segment description tools	326
11.8	Ink segment description tools.....	330
11.9	Video editing segment description tools	353
11.10	Structural relation classification schemes	381
12	Semantics description tools.....	393
12.1	Introduction	393
12.2	Abstraction model	396
12.3	Semantic entity description tools	397
12.4	Semantic attribute description tools	426
12.5	Semantic relation classification schemes	432
13	Navigation and access tools	449
13.1	Introduction	449
13.2	Summarization	449
13.3	Views, partitions and decompositions	482
13.4	Variations of the content.....	523
14	Content organization tools	530
14.1	Introduction	530
14.2	Collections.....	531
14.3	Models.....	545
14.4	Probability models.....	546
14.5	Analytic models	572
14.6	Cluster models.....	582
14.7	Classification models.....	584
15	User interaction tools.....	589
15.1	Introduction	589
15.2	User preferences.....	590
15.3	Usage History.....	622
Annex A	(informative) XPath Datatype definition based on W3C XPath specification	632
Annex B	(normative) Classification Schemes	638
Annex C	(informative) Registration procedure.....	726
Annex D	(informative) Patent statements	729
Bibliography		730

Foreword

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

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

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

ISO/IEC 15938-5 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*

Annex B forms a normative part of this part of ISO/IEC 15938. Annexes A, C and D are for information only.

Introduction

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

The following are specified in this standard:

- **Description Schemes (DS)** describe entities or relationships pertaining to multimedia content. Description Schemes specify the structure and semantics of their components, which may be Description Schemes, Descriptors, or datatypes.
- **Descriptors (D)** describe features, attributes, or groups of attributes of multimedia content.
- **Datatypes** are the basic reusable datatypes employed by Description Schemes and Descriptors.
- **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.

This document specifies the Multimedia Description Schemes (MDS) part of the ISO/IEC 15938 standard. This document specifies the MDS description tools and provides informative examples of descriptions. The normative syntax of the MDS description tools is specified in this document using the Description Definition Language (DDL) and the normative semantics is specified using text.

Information technology — Multimedia content description interface —

Part 5: Multimedia description schemes

1 Scope

1.1 Organization of the document

This International Standard specifies a metadata system for describing multimedia content. This document specifies the Multimedia Description Schemes (MDS) description tools (Description Schemes, Descriptors, and datatypes) that comprise ISO/IEC 15938-5. The following set of subclauses are provided for each MDS description tool, where optional subclauses are indicated as (optional):

- Syntax: specifies the normative syntax of the description tool using Description Definition Language (DDL) (see ISO/IEC 15938-2).
- Binary syntax (optional): specifies the normative binary representation for the description tool, where relevant. If a binary syntax is not specified, then the binary coding method of ISO/IEC 15938-1 shall apply for the description tool.
- Semantic: specifies the normative semantics of the description tool and each of its components (attributes and elements).
- Informative examples (optional): provides informative examples that illustrate the use or instantiation of the description tool in creating descriptions.

1.2 Overview of Multimedia Description Schemes

The MDS description tools are organized on the basis of functionality as shown in Figure 1.

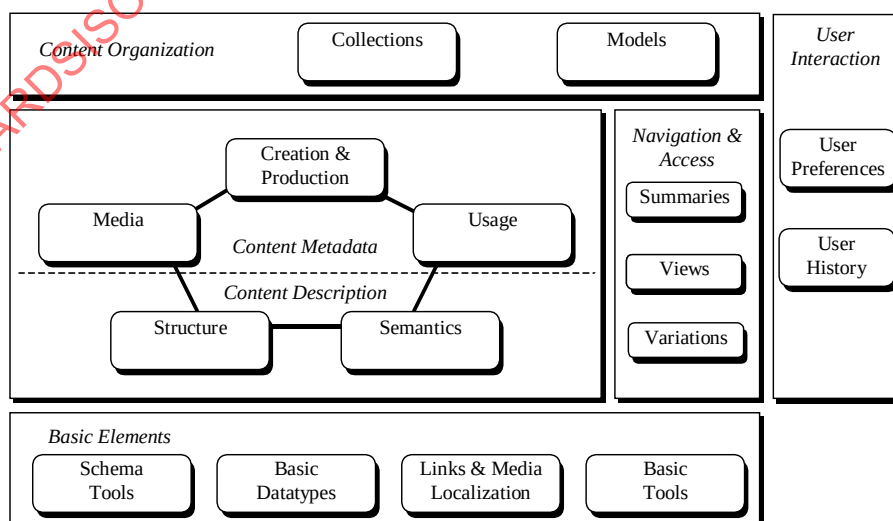


Figure 1 — Overview of the Multimedia Description Scheme (MDS) description tools

The basic elements, which are shown at the bottom level in Figure 1, form the building blocks for the higher-level description tools. The following basic elements are defined: schema tools (the root element, top-level types, description metadata, and packages), basic datatypes (integers, reals, vectors, matrices), linking and media localization tools (spatial and temporal localization), basic description tools (language, text, classification schemes). The basic elements are specified in the following clauses: clause 4 (Schema tools), clause 5 (Basic datatypes), clause 6 (Linking, identification and localization tools), and clause 7 (Basic description tools). Table 1 summarizes the basic elements.

Table 1 — Overview of basic elements

<i>Set of description tools</i>	<i>Functionality</i>
Schema tools (clause 4)	Specifies schema tools that facilitate the making of descriptions. The schema tools include the base type hierarchy of the description tools defined in ISO/IEC 15938, the root element and top-level tools, the multimedia content entity tools, and the package and description metadata tools.
Basic datatypes (clause 5)	Specifies the basic datatypes used by description tools defined in the ISO/IEC 15938.
Linking, identification and localization tools (clause 6)	Specifies the basic datatypes that are used for referencing within descriptions and linking of descriptions to multimedia content.
Basic description tools (clause 7)	Specifies basic tools that are used as components for building other description tools.

The content description tools, which are shown at the middle level in Figure 1, describe the features of the multimedia content and the immutable metadata related to the multimedia content. The following description tools for content description are defined: structure description tools (spatio-temporal segments of multimedia content) and semantic description tools (objects, events). The following description tools for content metadata are defined: media description (storage format, encoding), creation & production (title, creator, classification), and usage (access rights, publication). The content description and metadata tools are related in the sense that the content description tools use the content metadata tools. For example, a description of creation and production or media information can be attached to an individual video or video segment in order to describe the structure and creation and production of the multimedia content. Table 2 summarizes tools for describing content and related metadata.

Table 2 — Overview of tools for describing content and related metadata

<i>Set of description tools</i>	<i>Functionality</i>
Media description tools (clause 8)	Describes the storage of the multimedia data. The media features include the format, encoding, storage media. The tools allow multiple media description instances for the same multimedia content.
Creation and production description tools (clause 9)	Describes the creation and production of the multimedia content. The creation and production features include title, creator, classification, purpose of the creation, and so forth. The creation and production information is typically not extracted from the content but corresponds to metadata related to the content.
Usage description tools (clause 10)	Describes the usage of the multimedia content. The usage features include access rights, publication, and financial information. The usage information may change during the lifetime of the multimedia content.
Structure description tools (clause 11)	Describes the structure of the multimedia content. The structural features include spatial, temporal or spatio-temporal segments of the multimedia content.
Semantics description tools (clause 12)	Describes the "real-world" semantics related to or captured by the multimedia content. The semantic features include objects, events, concepts, and so forth.

The tools for navigation and access, which are shown at the middle level in Figure 1, describe the browsing, summarization, and access of content. These tools are specified in clause 13 (Navigation and access tools). The tools for content organization tools, shown at the top level in Figure 1, describe collections and models of multimedia content. These tools are specified in clause 14 (Content organization tools). The tools for user interaction, shown on the right in Figure 1, describe user preferences pertaining to consumption of multimedia and usage history. These tools are specified in clause 15 (User interaction tools). Table 3 summarizes tools for describing content organization, navigation and access, and user interaction.

Table 3 — Overview of tools for describing content organization, navigation and access, and user interaction

<i>Set of description tools</i>	<i>Functionality</i>
Navigation and access tools (clause 13)	Specifies tools that describe the navigation and access of multimedia. The navigation and access tools describe multimedia summaries and abstracts; describe partitions, views and decompositions of image, video, and audio signals in space, time and frequency; and describe the relationships between different variations of multimedia content.
Content organization tools (clause 14)	Specifies tools that describe the organization and modeling of multimedia content. The content organization tools include collections, probability models, analytic models, cluster models, and classification models.
User interaction tools (clause 15)	Specifies tools related to user interaction with multimedia content. The user interaction tools describe user preferences pertaining to multimedia content, and describe the usage history of users of multimedia content.

Finally, classification schemes, which organize terms that are used by the description tools, are specified in Annex B (Classification Schemes).

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 15938. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 15938 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau maintains a list of currently valid ITU-T Recommendations.

- ISO 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*
- ISO 639 (all parts), *Codes for the representation of names of languages*
- ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*
- ISO 3166-2, *Codes for the representation of names of countries and their subdivisions — Part 2: Country subdivision code*

Note: The current list of valid ISO 3166-1 country and ISO 3166-2 region codes is maintained by the ISO 3166 maintenance agency at the ISO Central Secretariat. Information on the current list of valid region and country codes can be found at <http://www.iso.org/mara/iso3166>.

- ISO 4217, *Codes for the representation of currencies and funds*

Note: The current list of valid ISO 4217 currency codes is maintained by the official maintenance authority British Standards Institution (<http://www.bsi-global.com/iso4217currency>).

- XML, *Extensible Markup Language (XML) 1.0*, 6 October 2000 (<http://www.w3.org/TR/2000/REC-xml-20001006>)
- XML Schema, *W3C Recommendation*, 2 May 2001 (<http://www.w3.org/XML/Schema>)
- XML Schema Part 0: *Primer*, W3C Recommendation, 2 May 2001 (<http://www.w3.org/TR/xmlschema-0/>)
- XML Schema Part 1: *Structures*, W3C Recommendation, 2 May 2001 (<http://www.w3.org/TR/xmlschema-1/>)
- XML Schema Part 2: *Datatypes*, W3C Recommendation, 2 May 2001 (<http://www.w3.org/TR/xmlschema-2/>)
- XPath, *XML Path Language*, W3C Recommendation, 16 November 1999 (<http://www.w3.org/TR/1999/REC-xpath-19991116>)

Note: These documents are maintained by the W3C (<http://www.w3.org>).

- RFC 2045 *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*
- RFC 2046 *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*
- RFC 2048, *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*
- RFC 2045-CHARSETS, *Registered Character set codes of RFC2045*
- RFC 2046-MIMETYPES, *Registered Mimetypes of RFC2046*

Note: The relevant lists can be obtained as follows:

- o MIMETYPES. The current list of registered mimetypes, as defined in RFC2046, RFC2048, is maintained by IANA (Internet Assigned Numbers Authority). It is available from <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/media-types/>.
- o CHARSETS. The current list of registered character set codes, as defined in RFC2045 and RFC2048 is maintained by IANA (Internet Assigned Numbers Authority). It is available from <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>.

3 Terms and definitions

3.1 Conventions

3.1.1 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.

3.1.2 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).

3.1.3 Documentation convention

The syntax of each description tool is specified using the constructs provided by the DDL specified in ISO/IEC 15938-2, and is presented in this document using a specific font and background as shown in the following example:

```
<complexType name="ExampleType">
  <sequence>
    <element name="Element1" type="string"/>
  </sequence>
  <attribute name="attribute1" type="string" default="attrvalue1"/>
</complexType>
```

The semantics of each description tool is specified in text using a table format, where each row contains the name and a definition of a type, element or attribute as shown in the following example:

Name	Definition
ExampleType	Specifies an ...
element1	Describes the ...
attribute1	Describes the ...

Diagrammatic notation is sometimes used to depict overviews of the description tools. Such overview diagrams generally depict aggregation and generalization relationships between description tools or the entities being described. In these diagrams, rectangular shapes containing a name denote description tools (DSs or Ds) that are used to describe entities. In some diagrams, multiple description tools are listed in a single rectangle. Large diamond shapes containing a name denote description tools (DSs or Ds) that are used to describe relationships. Furthermore, paths between rectangles or diamonds denote association, generalization or aggregation relationships. Generalization relationships are indicated by a solid path with a filled triangle pointing at the more

general entity. Aggregation (composition) relationships are indicated by a solid path with a (filled) diamond at the aggregating (composing) entity. Aggregation relationship paths may be accompanied by an indication of the multiplicity (minOccurs, maxOccurs) of the relation in text form.

The informative examples are included in separate subclauses, and are presented in this document using a separate font and background as shown in the following example:

```
<Example attribute1="example attribute value">
  <Element1>example element content</Element1>
</Example>
```

Moreover, the schema or set of description tools defined in this document follows a type-centric approach. The description tools are specified by defining the complexType or simpleType for each tool, but not declaring an element of this type at the global scope with the exception of the root element. However, in order to illustrate informative example descriptions in this document, the examples assume that an element of the example type (complexType or simpleType) has been declared, such as a member of another complexType or simpleType. For example, the description above assumes that the following declaration has been made:

```
<element name="Example" type="mpeg7:ExampleType">
```

The term "reserved" is used in specifying the semantics of some description tools. The term "reserved" indicates that particular values are reserved for use in future extensions of ISO/IEC 15938.

3.2 Wrapper of the schema

The description examples and syntax of description tools specified in this document assume that a schema wrapper is provided which identifies the XML Schema namespace (XML Schema) and MPEG-7 namespace:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  targetNamespace="urn:mpeg:mpeg7:schema:2001"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
<!-- Multimedia Description Schemes ISO/IEC 15938-5 -->
```

The following tag is used to close the schema:

```
</schema>
```

3.3 Terminology

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

3.3.1 Schema-related terminology

3.3.1.1

Attribute

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

3.3.1.2

Base type

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

3.3.1.3**Datatype**

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

3.3.1.4**Derived type**

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

3.3.1.5**Description**

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

3.3.1.6**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**.

3.3.1.7**Description Tool**

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

3.3.1.8**Descriptor**

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

3.3.1.9**Instantiation**

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

3.3.1.10**Element**

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

3.3.1.11**Schema**

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

3.3.1.12**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.

3.3.2 Content-related terminology**3.3.2.1****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**.

3.3.2.2**Acquisition**

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

3.3.2.3**Action**

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

3.3.2.4

Agent

A person, organization, or group of persons.

3.3.2.5

Audio

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

3.3.2.6

Audio-visual

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

3.3.2.7

Automatic

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

3.3.2.8

Classification Scheme

A list of defined terms and their meanings.

3.3.2.9

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**.

3.3.2.10

Copyright

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

3.3.2.11

Data

Essence

Multimedia Data

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

3.3.2.12

Editing

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

3.3.2.13

Entity

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

3.3.2.14

Event

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

3.3.2.15

Feature

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

3.3.2.16

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.

3.3.2.17**Format**

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

3.3.2.18**Frame**

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

3.3.2.19**Image**

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

3.3.2.20**Key frame**

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

3.3.2.21**Locator**

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

3.3.2.22**Model**

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

3.3.2.23**Manual**

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

3.3.2.24**Metadata**

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

3.3.2.25**Multimedia**

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

3.3.2.26**Navigation**

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

3.3.2.27**Object**

An object with a physical representation in the natural world.

3.3.2.28**Region**

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

3.3.2.29**Relation**

Any association among entities.

3.3.2.30**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.

3.3.2.31

Scene

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

3.3.2.32

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.

3.3.2.33

Segment

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

3.3.2.34

Semantics

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

3.3.2.35

Summary

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

3.3.2.36

User

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

3.3.2.37

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.

3.3.2.38

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.

3.3.2.39

Variation

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

3.3.2.40

Video

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

3.3.2.41

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.

3.3.2.42

Visual

Related to the visual modality.

3.3.2.43

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.

3.4 Symbols and abbreviated terms

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

AV:	Audio-visual
CS:	Classification Scheme
D:	Descriptor
Ds:	Descriptors
DCT:	Discrete Cosine Transform
DDL:	Description Definition Language
DS:	Description Scheme
DSs:	Description Schemes
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 Schemes
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
QCIF:	Quarter Common Intermediate Format
SMPTE:	Society of Motion Picture and Television Engineers
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

4 Schema tools

4.1 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 4 — Overview of Schema Tools

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

4.2 Base types

4.2.1 Introduction

This subclause specifies the base types of the description tools, as shown in Figure 2.

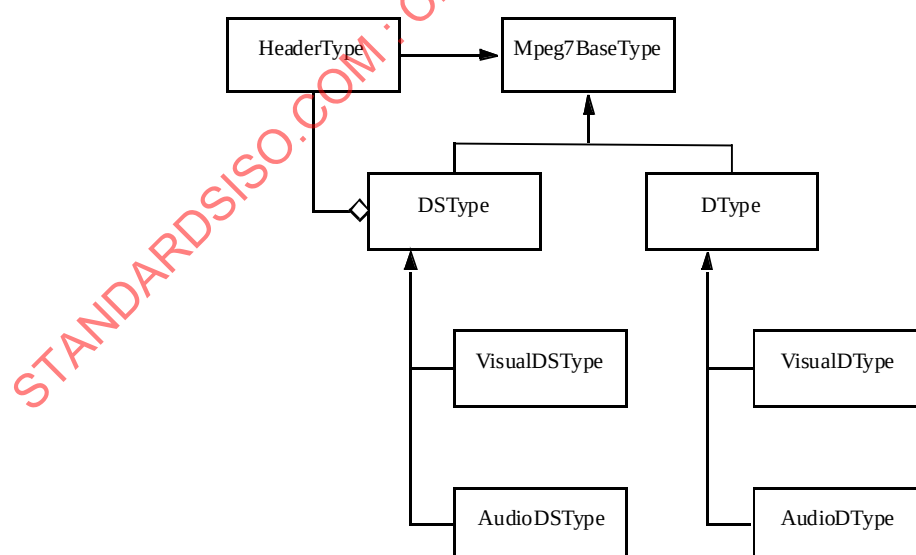


Figure 2 — Organization of base types of Mpeg7BaseType, DType, DSType, and HeaderType

The following base types are defined in this subclause:

- **Mpeg7BaseType**: provides the base abstract type of the type hierarchy. The **DSType**, **DType** and **HeaderType** extend **Mpeg7BaseType**.
 - o **DType**: extends **Mpeg7BaseType** and provides a base abstract type for Descriptors. The audio and visual Descriptors extend **DType**.
 - **VisualDType**: extends **DType** and provides a base abstract type for visual Descriptors. Descriptors that describe visual features of multimedia content extend **VisualDType**.
 - **AudioDType**: extends **DType** and provides a base abstract type for audio Descriptors. Descriptors that describe audio features of multimedia content extend **AudioDType**.
 - o **DSType**: extends **Mpeg7BaseType** and provides a base abstract type for Description Schemes. The audio and visual Description Schemes extend **DSType**. The **DSType** contains an element of type **HeaderType**.
 - **VisualDSType**: extends **DSType** and provides a base abstract type for visual Description Schemes. Description Schemes that describe visual concepts and visual features of multimedia content extend **VisualDSType**.
 - **AudioDSType**: extends **DSType** and provides a base abstract type for audio Description Schemes. Description Schemes that describe audio concepts and audio features extend **AudioDSType**.
 - o **HeaderType**: extends **Mpeg7BaseType** and provides a base abstract type for headers. A header collects information that recurs in a description in order to allow referencing using a reference tool.

4.2.2 Base types syntax

```

<!-- ##### -->
<!-- Definition of MPEG-7 Base types (4.2) -->
<!-- ##### -->

<!-- Definition of MPEG-7 base type -->
<complexType name="Mpeg7BaseType" abstract="true">
  <complexContent>
    <restriction base="anyType"/>
  </complexContent>
</complexType>

<!-- Definition of generic D -->
<complexType name="DType" abstract="true">
  <complexContent>
    <extension base="mpeg7:Mpeg7BaseType"/>
  </complexContent>
</complexType>

<!-- Definition of generic DS -->
<complexType name="DSType" abstract="true">
  <complexContent>
    <extension base="mpeg7:Mpeg7BaseType">
      <sequence>
        <element name="Header" type="mpeg7:HeaderType" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
      <attribute name="id" type="ID" use="optional"/>
      <attributeGroup ref="mpeg7:timePropertyGrp"/>
      <attributeGroup ref="mpeg7:mediaTimePropertyGrp"/>
    </extension>
  </complexContent>
</complexType>

```

```

    </complexContent>
</complexType>

<!-- Definition of Visual D -->
<complexType name="VisualDType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DType"/>
  </complexContent>
</complexType>

<!-- Definition of Audio D -->
<complexType name="AudioDType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DType"/>
  </complexContent>
</complexType>

<!-- Definition of Visual DS -->
<complexType name="VisualDSType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType"/>
  </complexContent>
</complexType>

<!-- Definition of Audio DS -->
<complexType name="AudioDSType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType"/>
  </complexContent>
</complexType>

<!-- Definition of Header datatype -->
<complexType name="HeaderType" abstract="true">
  <complexContent>
    <extension base="mpeg7:Mpeg7BaseType">
      <attribute name="id" type="ID" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

4.2.3 Base types semantics

Semantics of the Mpeg7BaseType:

Name	Definition
Mpeg7BaseType	Provides the topmost type of the base type hierarchy (abstract). The DSType, DType, and HeaderType extend Mpeg7BaseType.

Semantics of the DType:

Name	Definition
DType	Describes a feature of multimedia content. DType extends Mpeg7BaseType and provides a base abstract type for Descriptors. The AudioDType and VisualDType extend DType.

Semantics of the DType:

<i>Name</i>	<i>Definition</i>
DType	Describes entities or relationships related to the multimedia content. DType extends Mpeg7BaseType and provides a base abstract type for Description Schemes. The audio and visual Description Schemes extend DType. The DType contains an element of type HeaderType.
Header	Describes the header of a description (optional). Header collects information that recurs in a description in order to allow referencing using a reference tool.
id	Identifies the id of a multimedia entity or relationship (optional). The id shall be used for referencing and relations in order to connect descriptions.
timePropertyGrp	Describes the properties of a time line associated with AV content (optional). timePropertyGrp is defined in 6.4.8.
mediaTimePropertyGrp	Describes the properties of a time line associated with AV media (optional). mediaTimePropertyGrp is defined in 6.4.16.

Semantics of the VisualDType:

<i>Name</i>	<i>Definition</i>
VisualDType	Describes visual features of multimedia content (abstract). VisualDType extends DType and provides a base abstract type for visual Descriptors.

Semantics of the AudioDType:

<i>Name</i>	<i>Definition</i>
AudioDType	Describes audio features of multimedia content (abstract). AudioDType extends DType and provides a base abstract type for audio Descriptors.

Semantics of the VisualDSType:

<i>Name</i>	<i>Definition</i>
VisualDSType	Describes visual concepts and visual features of multimedia content (abstract). VisualDSType extends DType and provides a base abstract type for visual Description Schemes.

Semantics of the AudioDSType:

<i>Name</i>	<i>Definition</i>
AudioDSType	Describes audio concepts and audio features of multimedia content (abstract). AudioDSType extends DType and provides a base abstract type for audio Description Schemes.

Semantics of the HeaderType:

Name	Definition
HeaderType	Describes information that recurs in a description in order to allow referencing using a reference tool (abstract). HeaderType extends Mpeg7BaseType and provides a base abstract type for headers.
id	Identifies the id of the HeaderType (optional). The id shall be used for referencing (see 6.2) and relations (see 7.7) in order to reuse the information in the header in different parts of a description.

4.3 Root element

4.3.1 Introduction

This subclause specifies the root type and the root element. The root element shall be used as the topmost element in a description. The root type provides metadata about the description as well as information that is common to the description, such as the language of the text and the convention for specifying time. The root element provides a choice of elements for creating either a complete description or a description unit, which are defined as follows:

- **Complete Description:** describes multimedia content using the top-level types. For example, the description of an image is a complete description.
- **Description Unit:** describes an instance of a D, DS, or header. A description unit can be used to represent partial information from a complete description. For example, the description of a shape or color is a description unit.

4.3.2 Root element syntax

```
<!-- ##### -->
<!-- Definition of MPEG-7 root element (4.3) -->
<!-- ##### -->

<!-- Definition of Mpeg7 Type-->
<complexType name="Mpeg7Type" abstract="true">
  <sequence>
    <element name="DescriptionMetadata" type="mpeg7:DescriptionMetadataType"
      minOccurs="0"/>
  </sequence>
  <attribute ref="xml:lang" use="optional"/>
  <attributeGroup ref="mpeg7:timePropertyGrp"/>
  <attributeGroup ref="mpeg7:mediaTimePropertyGrp"/>
</complexType>

<!-- Definition of Mpeg7 Element -->
<element name="Mpeg7">
  <complexType>
    <complexContent>
      <extension base="mpeg7:Mpeg7Type">
        <choice>
          <element name="DescriptionUnit" type="mpeg7:Mpeg7BaseType"/>
          <element name="Description" type="mpeg7:CompleteDescriptionType"
            minOccurs="1" maxOccurs="unbounded"/>
        </choice>
      </extension>
    </complexContent>
  </complexType>
</element>
```

4.3.3 Root element semantics

Semantics of the Mpeg7Type:

<i>Name</i>	<i>Definition</i>
Mpeg7Type	The type of the root element.
DescriptionMetadata	Describes the metadata for the descriptions contained within the instance of the root type (optional). The description metadata applies to the elements contained within the instance of the root element unless new description metadata is described for those elements. DescriptionMetadataType is defined in 4.5.3.
xml:lang	Identifies the language of the textual content in the description (optional). If not specified, the language is unknown. xml:lang is defined in (XML).
timePropertyGrp	Describes the properties of a time line associated with AV content (optional). timePropertyGrp is defined in 6.4.8.
mediaTimePropertyGrp	Describes the properties of a time line associated with AV media (optional). mediaTimePropertyGrp is defined in 6.4.16.

Semantics of the Mpeg7 root element:

<i>Name</i>	<i>Definition</i>
Mpeg7	Serves as the root element of the description. Mpeg7 shall be used as the topmost element in a description.
DescriptionUnit	Describes an instance of one of a D, DS, or header (optional). A description unit can be used to represent partial information from a complete description. For example, DescriptionUnit can be used to describe a shape using the RegionShape D (defined in ISO/IEC 15938-3).
Description	Describes multimedia content using the top-level types (optional). For example, Description can be used to describe an image using ContentEntityType, where MultimediaContent is of type ImageType.

4.3.4 Root element examples (informative)

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 type="UUID">
      098f2470-bae0-11cd-b579-08002b30bfeb
    </PublicIdentifier>
    <PrivateIdentifier>completeDescriptionExample</PrivateIdentifier>
    <Creator>
      <Role href="creatorCS"><Name>Creator</Name></Role>
      <Agent xsi:type="PersonType">
        <Name>
          <GivenName>Yoshiaki</GivenName>
          <FamilyName>Shibata</FamilyName>
        </Name>
      </Agent>
    </Creator>
    <CreationLocation>
      <Region>jp</Region>
      <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>
      <RightsID>RID#</RightsID>
    </Rights>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <!-- more elements here -->
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```


4.4 Top-level types

4.4.1 Introduction

This subclause specifies the top-level types. The top-level types are used in complete descriptions to describe multimedia content and metadata related to content management. Each top-level type contains the description tools that are relevant for a particular description task, i.e., describing an image or a video. The following top-level types are defined in this subclause:

- CompleteDescriptionType (abstract): top-level type for complete descriptions.
 - o ContentDescriptionType (abstract): top-level type for complete descriptions of multimedia content.
 - ContentEntityType: top-level type for describing multimedia content entities such as images, videos, audio, collections, and so forth.
 - ContentAbstractionType (abstract): top-level type for describing abstractions of multimedia content:
 - SemanticDescriptionType: top-level type for describing semantics of multimedia content.
 - ModelDescriptionType: top-level type for describing models of multimedia content.
 - SummaryDescriptionType: top-level type for describing summaries of multimedia content.
 - ViewDescriptionType: top-level type for describing views and view decompositions of audio-visual signals.
 - VariationDescriptionType: top-level type for describing variations of multimedia content.
 - o ContentManagementType (abstract): top-level type for describing metadata related to content management:
 - UserDescriptionType: top-level type for describing a user of a multimedia system.
 - MediaDescriptionType: top-level type for describing the media information of multimedia content.
 - CreationDescriptionType: top-level type for describing the process of creating multimedia content.
 - UsageDescriptionType: top-level type for describing the usage of multimedia content.
 - ClassificationSchemeDescriptionType: top-level type for describing a classification scheme for multimedia content.

The top-level types are organized under the type hierarchy shown in Figure 3. The CompleteDescriptionType forms the root base type of the hierarchy. The top-level types ContentDescriptionType and ContentManagementType extend CompleteDescriptionType. The top-level types ContentEntityType and ContentAbstractionType extend ContentDescriptionType.

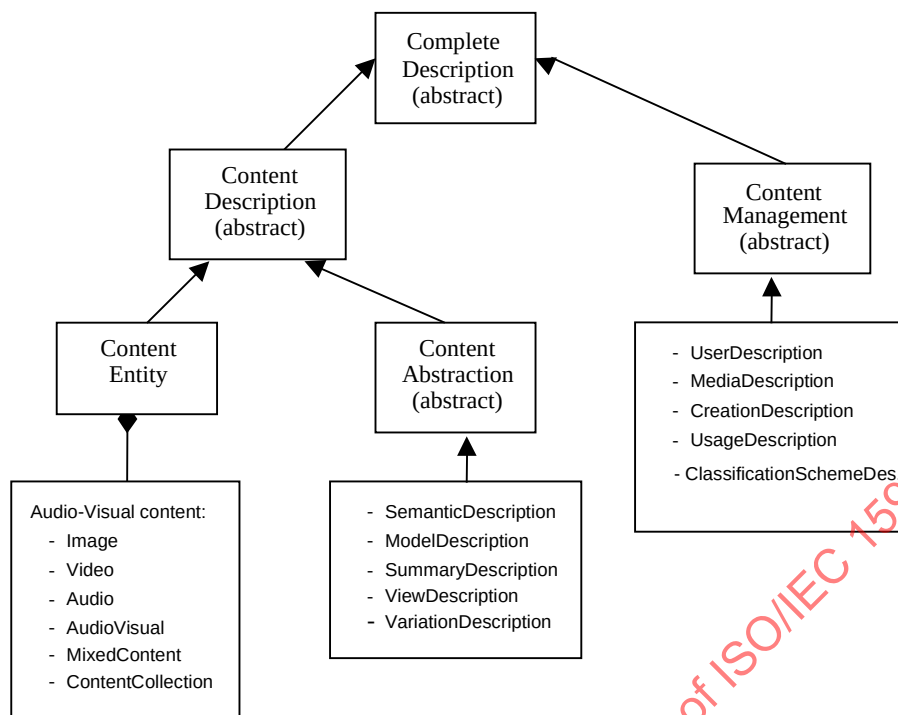


Figure 3 — Illustration of the type derivation hierarchy for top-level types

4.4.2 Complete description types

4.4.2.1 Introduction

The complete description types describe multimedia content, abstractions of multimedia content, and metadata related to the management of multimedia content. CompleteDescriptionType is the base type for ContentDescriptionType and ContentManagementType. Since the root element allows the instantiation of multiple complete description top-level types in a single description, the CompleteDescriptionType includes a Relationships element of type GraphType that describes the relationships among the DS instances in the description. The CompleteDescriptionType includes description metadata that describes the metadata for the descriptions contained within the instance of the complete description top-level type.

4.4.2.2 Complete description type syntax

```

<!-- ##### -->
<!-- Definition of Complete description top-level types (4.4.2) -->
<!-- ##### -->

<!-- Definition of CompleteDescription Top-level Type -->
<complexType name="CompleteDescriptionType" abstract="true">
  <sequence>
    <element name="DescriptionMetadata"
      type="mpeg7:DescriptionMetadataType" minOccurs="0"/>
    <element name="Relationships" type="mpeg7:GraphType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="OrderingKey" type="mpeg7:OrderingKeyType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

```

4.4.2.3 Complete description type semantics

Semantics of the CompleteDescriptionType:

Name	Definition
CompleteDescriptionType	Describes multimedia content, abstractions of multimedia content, and management of multimedia content (abstract). CompleteDescriptionType is the base type for ContentDescriptionType and ContentManagementType.
DescriptionMetadata	Describes the metadata for the descriptions contained within the instance of the top-level type (optional). The description metadata applies to the elements contained within the instance of the root element unless new description metadata is described for those elements. DescriptionMetadataType is defined in 4.5.3.
Relationships	Describes a graph of the relationships among the DS instances in the description (optional). Also describes relationships across different individual sub-descriptions that appear under a single root element. GraphType is defined in 7.7.2.
OrderingKey	Describes an ordering of instances within a description (optional). The ordering does not constrain the actual order of appearance within the description itself. OrderingKeyType is defined in 7.8.2.

4.4.3 Content description types

4.4.3.1 Introduction

The content description top-level types describe multimedia content entities and abstractions of multimedia content.

4.4.3.2 Content description types syntax

```

<!-- ##### -->
<!-- Definition of Content description top-level types (4.4.3) -->
<!-- ##### -->
<!-- Definition of ContentDescription Top-level Type -->
<complexType name="ContentDescriptionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:CompleteDescriptionType">
      <sequence>
        <element name="Affective" type="mpeg7:AffectiveType" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of ContentEntity Top-level Type -->
<complexType name="ContentEntityType">
  <complexContent>
    <extension base="mpeg7:ContentDescriptionType">
      <sequence>
        <element name="MultimediaContent" type="mpeg7:MultimediaContentType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- Definition of ContentAbstraction Top-level Type -->
  <complexType name="ContentAbstractionType" abstract="true">
    <complexContent>
      <extension base="mpeg7:ContentDescriptionType"/>
    </complexContent>
  </complexType>

  <!-- Definition of SemanticDescription Top-level Type -->
  <complexType name="SemanticDescriptionType">
    <complexContent>
      <extension base="mpeg7:ContentAbstractionType">
        <choice>
          <element name="Semantics" type="mpeg7:SemanticType" minOccurs="1"
            maxOccurs="unbounded"/>
          <element name="ConceptCollection" type="mpeg7:ConceptCollectionType"
            minOccurs="1" maxOccurs="unbounded"/>
        </choice>
      </extension>
    </complexContent>
  </complexType>

  <!-- Definition of ModelDescription Top-level Type -->
  <complexType name="ModelDescriptionType">
    <complexContent>
      <extension base="mpeg7:ContentAbstractionType">
        <sequence>
          <element name="Model" type="mpeg7:ModelType" minOccurs="1"
            maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- Definition of SummaryDescription Top-level Type -->
  <complexType name="SummaryDescriptionType">
    <complexContent>
      <extension base="mpeg7:ContentAbstractionType">
        <sequence>
          <element name="Summarization" type="mpeg7:SummarizationType"
            minOccurs="1" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- Definition of ViewDescription Top-level Type -->
  <complexType name="ViewDescriptionType">
    <complexContent>
      <extension base="mpeg7:ContentAbstractionType">
        <choice>
          <element name="View" type="mpeg7:ViewType" minOccurs="1"
            maxOccurs="unbounded"/>
          <element name="ViewSet" type="mpeg7:ViewSetType" minOccurs="1"
            maxOccurs="unbounded"/>
          <element name="ViewDecomposition" type="mpeg7:ViewDecompositionType"

```

```

        minOccurs="1" maxOccurs="unbounded"/>
    </choice>
</extension>
</complexContent>
</complexType>

<!-- Definition of VariationDescription Top-level Type -->
<complexType name="VariationDescriptionType">
    <complexContent>
        <extension base="mpeg7:ContentAbstractionType">
            <choice>
                <element name="VariationSet" type="mpeg7:VariationSetType"
                    minOccurs="1" maxOccurs="unbounded"/>
                <element name="Variation" type="mpeg7:VariationType" minOccurs="1"
                    maxOccurs="unbounded"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

```

4.4.3.3 Content description types semantics

Semantics of the ContentDescriptionType:

Name	Definition
ContentDescriptionType	Top-level type for describing multimedia content and abstractions of multimedia content (abstract). Provides the base type of ContentEntityType and ContentAbstractionType. ContentDescriptionType extends CompleteDescriptionType.
Affective	Describes the affective response of users to the multimedia content (optional). The AffectiveType is defined in 7.9.

Semantics of the ContentEntityType:

Name	Definition
ContentEntityType	Top-level type for describing multimedia content entities such as images, video, audio, and so forth. ContentEntityType extends ContentDescriptionType.
MultimediaContent	Describes an instance of multimedia content.

Semantics of the ContentAbstractionType:

Name	Definition
ContentAbstractionType	Top-level type for describing abstractions of multimedia content (abstract). An abstraction is a secondary content representation that is created from or is related to the multimedia content. For example, a summary of video or a model of audio-visual features is an abstraction of the multimedia content. ContentAbstractionType extends ContentDescriptionType.

Semantics of the SemanticDescriptionType:

<i>Name</i>	<i>Definition</i>
SemanticDescriptionType	Top-level type for describing semantics of multimedia content including real-life events, objects, concepts, and narratives. SemanticDescriptionType extends ContentAbstractionType.
Semantics	Describes the semantics of multimedia content (optional). SemanticType is defined in 12.3.4.
ConceptCollection	Describes a collection of semantic concepts related to multimedia content (optional). ConceptCollectionType is defined in 14.2.5.

Semantics of the ModelDescriptionType:

<i>Name</i>	<i>Definition</i>
ModelDescriptionType	Top-level type for describing models of multimedia content. ModelDescriptionType extends ContentAbstractionType.
Model	Describes a model of multimedia content. ModelType is defined in 14.3.

Semantics of the SummaryDescriptionType:

<i>Name</i>	<i>Definition</i>
SummaryDescriptionType	Top-level type for describing summaries of AV content. SummaryDescriptionType extends ContentAbstractionType.
Summarization	Describes a summarization of AV content. SummarizationType is defined in 13.2.

Semantics of the ViewDescriptionType:

<i>Name</i>	<i>Definition</i>
ViewDescriptionType	Top-level type for describing views, sets of views, and view decompositions of image, video, or audio signal. ViewDescriptionType extends ContentAbstractionType.
View	Describes a view of an image, video, or audio signal (optional). ViewType is defined in 13.3.5
ViewSet	Describes a set of views of an image, video, or audio signal (optional). ViewSetType is defined in 13.3.12.
ViewDecomposition	Describes a view decomposition of an image, video, or audio signal (optional). ViewDecompositionType is defined in 13.3.11.

Semantics of the VariationDescriptionType:

Name	Definition
VariationDescriptionType	Top-level type for describing variations of multimedia content. VariationDescriptionType extends ContentAbstractionType.
VariationSet	Describes a set of variations of multimedia content (optional). VariationSetType is defined in 13.4.2.
Variation	Describes a variation of multimedia content (optional). VariationType is defined in 13.4.2.

4.4.3.4 Content description types examples (informative)

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 12.3.4.

```
<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 14.5.3.

```

<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 13.2.4.

```

<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 13.3.6.

```
<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 13.4.2.

```
<Mpeg7>
  <Description xsi:type="VariationDescriptionType">
    <VariationSet>
      <Source xsi:type="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>
```

4.4.4 Content management types

4.4.4.1 Introduction

This subclause specifies top-level types that support the management of multimedia content.

4.4.4.2 Content management types syntax

```

<!-- ##### -->
<!-- Definition of Content management top-level types (4.4.4) -->
<!-- ##### -->

<!-- Definition of ContentManagement Top-level Type -->
<complexType name="ContentManagementType" abstract="true">
  <complexContent>
    <extension base="mpeg7:CompleteDescriptionType"/>
  </complexContent>
</complexType>

<!-- Definition of UserDescription Top-level Type -->
<complexType name="UserDescriptionType">
  <complexContent>
    <extension base="mpeg7:ContentManagementType">
      <sequence>
        <element name="User" type="mpeg7:AgentType"
          minOccurs="0"/>
        <element name="UserPreferences" type="mpeg7:UserPreferencesType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="UsageHistory" type="mpeg7:UsageHistoryType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of MediaDescription Top-level Type -->
<complexType name="MediaDescriptionType">
  <complexContent>
    <extension base="mpeg7:ContentManagementType">
      <sequence>
        <element name="MediaInformation"
          type="mpeg7:MediaInformationType" minOccurs="1"
          maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of CreationDescription Top-level Type -->
<complexType name="CreationDescriptionType">
  <complexContent>
    <extension base="mpeg7:ContentManagementType">
      <sequence>
        <element name="CreationInformation"
          type="mpeg7:CreationInformationType" minOccurs="1"
          maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>

```

```

</complexType>

<!-- Definition of UsageDescription Top-level Type -->
<complexType name="UsageDescriptionType">
  <complexContent>
    <extension base="mpeg7:ContentManagementType">
      <sequence>
        <element name="UsageInformation" type="mpeg7:UsageInformationType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of ClassificationSchemeDescription Top-level Type -->
<complexType name="ClassificationSchemeDescriptionType">
  <complexContent>
    <extension base="mpeg7:ContentManagementType">
      <choice>
        <element name="ClassificationScheme"
          type="mpeg7:ClassificationSchemeType" minOccurs="1"
          maxOccurs="unbounded"/>
        <element name="ClassificationSchemeBase"
          type="mpeg7:ClassificationSchemeBaseType" minOccurs="1"
          maxOccurs="unbounded"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

4.4.4.3 Content management types semantics

Semantics of the ContentManagementType:

Name	Definition
ContentManagementType	Top-level type for describing metadata related to the management of multimedia content (abstract). ContentManagementType extends CompleteDescriptionType.

Semantics of the UserDescriptionType:

Name	Definition
UserDescriptionType	Top-level type for describing users of multimedia systems. UserDescriptionType extends ContentManagementType.
User	Describes a user of a multimedia system (optional). A user can be a person, a group of persons, or an organization. User if of type AgentType, which is defined in 7.5.
UserPreferences	Describes the preferences of the user of a multimedia system (optional). UserPreferencesType is defined in 15.2.
UsageHistory	Describes the usage history of the user of a multimedia system (optional). UsageHistoryType is defined in 15.3.

Semantics of the MediaDescriptionType:

<i>Name</i>	<i>Definition</i>
MediaDescriptionType	Top-level type for describing the media information of multimedia content. MediaDescriptionType extends ContentManagementType.
MediaInformation	Describes the media information of multimedia content. MediaInformationType is defined in 8.2.

Semantics of the CreationDescriptionType:

<i>Name</i>	<i>Definition</i>
CreationDescriptionType	Top-level type for describing the processes of creating multimedia content. CreationDescriptionType extends ContentManagementType.
CreationInformation	Describes the information about the creation of multimedia content. CreationInformationType is defined in 9.2.

Semantics of the UsageDescriptionType:

<i>Name</i>	<i>Definition</i>
UsageDescriptionType	Top-level type for describing the usage of multimedia content. UsageDescriptionType extends ContentManagementType.
UsageInformation	Describes usage rights of multimedia content. UsageInformationType is defined in 10.2.

Semantics of the ClassificationSchemeDescriptionType:

<i>Name</i>	<i>Definition</i>
ClassificationSchemeDescriptionType	Top-level type for describing classification schemes for multimedia content. ClassificationSchemeDescriptionType extends ContentManagementType.
ClassificationScheme	Describes a classification scheme or ontology for multimedia content. ClassificationSchemeType is defined in 7.4.2.
ClassificationSchemeBase	Describes a classification scheme or ontology for multimedia content using the base type for classification schemes. ClassificationSchemeBaseType is defined in 7.4.2.

4.4.4.4 Content management types example (informative)

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 15.2.3.

```

<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>
      <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 8.2.1.

```

<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 9.2.1.

```
<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 10.2.1.

```
<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 7.4.2.

```

<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>

```

4.4.5 Multimedia content entity description tools

4.4.5.1 Introduction

This subclause specifies tools for describing different multimedia content entities. The different multimedia content entity tools correspond to the different types of multimedia content: images, video, audio, AV data, mixed multimedia content, signals, ink content, and edited video.

4.4.5.2 Multimedia content entity description tools syntax

```

<!-- ##### -->
<!-- Definition of Multimedia content entity tools (4.4.5) -->
<!-- ##### -->

<!-- Definition of MultimediaContent Content Entity -->
<complexType name="MultimediaContentType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
    </extension>
  </complexContent>
</complexType>

<!-- Definition of Image Content Entity -->
<complexType name="ImageType">
  <complexContent>
    <extension base="mpeg7:MultimediaContentType">
      <sequence>
        <element name="Image" type="mpeg7:StillRegionType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of Video Content Entity -->
<complexType name="VideoType">
  <complexContent>
    <extension base="mpeg7:MultimediaContentType">
      <sequence>

```

```

        <element name="Video" type="mpeg7:VideoSegmentType"/>
    </sequence>
</extension>
</complexContent>
</complexType>

<!-- Definition of Audio Content Entity -->
<complexType name="AudioType">
    <complexContent>
        <extension base="mpeg7:MultimediaContentType">
            <sequence>
                <element name="Audio" type="mpeg7:AudioSegmentType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of AudioVisual Content Entity -->
<complexType name="AudioVisualType">
    <complexContent>
        <extension base="mpeg7:MultimediaContentType">
            <sequence>
                <element name="AudioVisual" type="mpeg7:AudioVisualSegmentType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of Multimedia Content Entity -->
<complexType name="MultimediaType">
    <complexContent>
        <extension base="mpeg7:MultimediaContentType">
            <sequence>
                <element name="Multimedia" type="mpeg7:MultimediaSegmentType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of MultimediaCollection Content Entity -->
<complexType name="MultimediaCollectionType">
    <complexContent>
        <extension base="mpeg7:MultimediaContentType">
            <choice>
                <element name="Collection" type="mpeg7:CollectionType" minOccurs="1"
                    maxOccurs="unbounded"/>
                <element name="StructuredCollection"
                    type="mpeg7:StructuredCollectionType" minOccurs="1"
                    maxOccurs="unbounded"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of Signal Content Entity -->
<complexType name="SignalType">
    <complexContent>
        <extension base="mpeg7:MultimediaContentType">
            <choice>
                <element name="ImageSignal" type="mpeg7:StillRegionType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

```



```

        <element name="VideoSignal" type="mpeg7:VideoSegmentType"/>
        <element name="AudioSignal" type="mpeg7:AudioSegmentType"/>
        <element name="SignalRef" type="mpeg7:ReferenceType"/>
    </choice>
</extension>
</complexContent>
</complexType>

<!-- Definition of InkContent Content Entity -->
<complexType name="InkContentType">
    <complexContent>
        <extension base="mpeg7:MultimediaContentType">
            <sequence>
                <element name="InkContent" type="mpeg7:InkSegmentType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of AnalyticEditedVideo Content Entity -->
<complexType name="AnalyticEditedVideoType">
    <complexContent>
        <extension base="mpeg7:MultimediaContentType">
            <sequence>
                <element name="AnalyticEditedVideo"
                    type="mpeg7:AnalyticEditedVideoSegmentType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

4.4.5.3 Multimedia content entity description tools semantics

Semantics of the MultimediaContentType:

Name	Definition
MultimediaContentType	Describes a multimedia content entity (abstract). The multimedia content entity shall correspond to one of a variety of multimedia content types including images, video, audio, content collections, ink content, and so forth. MultimediaContentType extends DSType.

Semantics of the ImageType:

<i>Name</i>	<i>Definition</i>
ImageType	Multimedia content entity for describing an image. An image refers to 2D spatially-varying visual data acquired from a visual source. ImageType extends MultimediaContentType.
Image	Describes the image. The image may refer to 2D spatial data or to a video frame. The image may correspond to a still region of an image or video frame. Image is of type StillRegionType, which is defined in 11.4.2.

Semantics of the VideoType:

<i>Name</i>	<i>Definition</i>
VideoType	Multimedia content entity for describing a video. Video refers to time-varying 2-D spatial data commonly represented as a temporally discrete sequence of images or video frames. VideoType extends MultimediaContentType.
Video	Describes the video. The video may be a full video sequence or a temporal segment, shot, or clip extracted from a video sequence. Video is of type VideoSegmentType, which is defined in 11.4.8.

Semantics of the AudioType:

<i>Name</i>	<i>Definition</i>
AudioType	Multimedia content entity for describing audio. Audio refers to time-varying 1-D data intended for listening or hearing. AudioType extends MultimediaContentType.
Audio	Describes the audio. The audio may be a full audio sequence or a temporal segment or clip extracted from an audio sequence. Audio is of type AudioSegmentType, which is defined in 11.5.2.

Semantics of the AudioVisualType:

<i>Name</i>	<i>Definition</i>
AudioVisualType	Multimedia content entity for describing AV data. AV refers to content consisting of both audio and video. AudioVisualType extends MultimediaContentType.
AudioVisual	Describes the AV data. The AV data may be a full AV sequence or a temporal segment or clip extracted from an AV sequence. AudioVisual is of type AudioVisualSegmentType, which is defined in 11.6.2.

Semantics of the MultimediaType:

<i>Name</i>	<i>Definition</i>
MultimediaType	Multimedia content entity for describing multimedia data. Multimedia refers to content comprising one or modalities or content types, such as images, audio, video, 3D models, ink content, and so forth. MultimediaType extends MultimediaContentType.
Multimedia	Describes the multimedia data. The multimedia data may be a composite of multimedia content or a single item of multimedia content that forms a multimedia presentation. Multimedia is of type MultimediaSegmentType, which is defined in 11.7.2.

Semantics of the MultimediaCollectionType:

<i>Name</i>	<i>Definition</i>
MultimediaCollectionType	Multimedia content entity for describing a collection of multimedia data. A multimedia collection is loose collection that does not necessarily form a single coherent presentation. The multimedia collection may contain multiple collections of content and segment. MultimediaCollectionType extends MultimediaContentType.
Collection	Describes a collection related to multimedia content. Examples include collections of multimedia content, collections of segments, and collections of descriptor instances. Each collection is an unordered, unbounded, nested set of items related to multimedia content. CollectionType is defined in 14.2.1.

Semantics of the SignalType:

<i>Name</i>	<i>Definition</i>
SignalType	Multimedia content entity for describing an image, video, or audio signal that can be represented as continuous or discrete sample data in space and/or time. For example, a signal can correspond to a discrete 1-D audio signal, a 2-D discrete spatial lattice of image data, or a time-varying video signal. These signal types of image, video, or audio signal data are distinguished from other types of multimedia content, such as complex multimedia presentations, in order to allow the description of views in space, time, and frequency. SignalType is used in ViewType for describing views, partitions, and decompositions of discrete lattice data. SignalType extends MultimediaContentType.
ImageSignal	Describes the image signal. ImageSignal is of type StillRegionType, which is defined in 11.4.2.
VideoSignal	Describes the video signal. VideoSignal is of type VideoSegmentType, which is defined in 11.4.8.
AudioSignal	Describes the audio signal. AudioSignal is of type AudioSegmentType, which is defined in 11.5.2.
SignalRef	Describes a reference to a description of an image, video, or audio signal. SignalRef is of type ReferenceType, which is defined in 6.2.1.

Semantics of the InkContentType:

Name	Definition
InkContentType	Multimedia content entity for describing ink content. Ink content refers to digital written or stroke data generated from various pen devices as well as ink applications (e.g., electronic whiteboard applications, form filling and ink messaging). InkContentType extends MultimediaContentType.
InkContent	Describes the ink content. The ink content may be a full ink document or a segment extracted from an ink document. InkContent is of type InkSegmentType, which is defined in 11.8.2.

Semantics of the AnalyticEditedVideoType:

Name	Definition
AnalyticEditedVideoType	Multimedia content entity for describing the video in terms of the analysis of its editing of clips and transitions. AnalyticEditedVideoType extends MultimediaContentType.
AnalyticEditedVideo	Describes the edited video. The analytic edited video may be a full edited video or a segment of edited video. AnalyticEditedVideo is of type AnalyticEditedVideoSegmentType, which is defined in 11.9.2.

4.4.5.4 Multimedia content entity description tools examples (informative)

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>

```

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>
```

```

        </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>
        <AnalyticEditingTemporalDecomposition>
          <Shot> <!-- ... --> </Shot>
        </AnalyticEditingTemporalDecomposition>
      </AnalyticEditedVideo>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

4.5 Description metadata tools

4.5.1 Introduction

This subclause specifies tools for describing metadata about descriptions. The description metadata includes packages of description tools and description metadata about the creation of descriptions.

4.5.2 Package DS

4.5.2.1 Introduction

The Package DS organizes and labels the description tools for ease of use and navigation of the description tools. The Package DS allows nesting of package descriptions within package descriptions.

4.5.2.2 Package DS syntax

```

<!-- ##### -->
<!-- Definition of Package DS (4.5.2) -->
<!-- ##### -->

<!-- Definition of Package DS -->
<complexType name="PackageType">
  <complexContent>

```



```

<extension base="mpeg7:DSType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element name="Scheme">
      <complexType>
        <attribute name="name" type="QName" use="required" />
      </complexType>
    </element>
    <element name="Package" type="mpeg7:PackageType" />
  </choice>
  <attribute name="name" type="string" use="required" />
</extension>
</complexContent>
</complexType>

```

4.5.2.3 Package DS semantics

Semantics of the PackageType:

Name	Definition
PackageType	Describes a package of description tools.
Scheme	Describes a description tool by giving its name.
name	Identifies the name of a description tool.
Package	Describes the nesting of package descriptions with the package (optional).
name	Identifies the name of the title of the package.

4.5.2.4 Package DS examples (informative)

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>
        </Package>
      </Package>
    </Package>
  </DescriptionMetadata>
</Mpeg7>

```

```

        </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>
</DescriptionMetadata>
<!-- more elements here -->
</Mpeg7>

```

4.5.3 DescriptionMetadata Header

4.5.3.1 Introduction

The DescriptionMetadata Header describes metadata concerning a description. Such metadata includes information identifying the description (privately or publicly) and describing the creation of the description, the version of the description, and the rights associated with the description.

4.5.3.2 DescriptionMetadata header syntax

```

<!-- ##### -->
<!-- Definition of DescriptionMetadata Header (4.5.3) -->
<!-- ##### -->

<!-- Definition of DescriptionMetadata Header -->
<complexType name="DescriptionMetadataType">
    <complexContent>
        <extension base="mpeg7:HeaderType">
            <sequence>
                <element name="Confidence" type="mpeg7:zeroToOneType" minOccurs="0"/>
                <element name="Version" type="string" minOccurs="0"/>
                <element name="LastUpdate" type="mpeg7:timePointType" minOccurs="0"/>
                <element name="Comment" type="mpeg7:TextAnnotationType" minOccurs="0"/>
                <element name="PublicIdentifier" type="mpeg7:UniqueIDType"
                    minOccurs="0" maxOccurs="unbounded"/>
                <element name="PrivateIdentifier" type="string"
                    minOccurs="0" maxOccurs="unbounded"/>
                <element name="Creator" type="mpeg7:CreatorType"
                    minOccurs="0" maxOccurs="unbounded"/>
                <element name="CreationLocation" type="mpeg7:PlaceType" minOccurs="0"/>
                <element name="CreationTime" type="mpeg7:timePointType" minOccurs="0"/>
                <element name="Instrument" type="mpeg7:CreationToolType"
                    minOccurs="0" maxOccurs="unbounded"/>
                <element name="Rights" type="mpeg7:RightsType" minOccurs="0"/>
                <element name="Package" type="mpeg7:PackageType" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

4.5.3.3 DescriptionMetadata header Semantics

Semantics of the DescriptionMetadataType.

Name	Definition
DescriptionMetadataType	Header describing metadata concerning a description including information identifying the description (privately or publicly), and describing the creation of the description, the version of the description, and the rights associated with the description. DescriptionMetadataType extends HeaderType.
Confidence	Indicates the confidence in the correctness of the description to which the description metadata is attached (optional). The value zero represents no confidence and the value 1.0 perfect confidence. All other values between zero and one represent intermediate levels of confidence.
Version	Specifies the version of the description to which the description metadata is attached (optional). The format for the version information is application dependent.
LastUpdated	Specifies the time when information in the description to which the description metadata is attached was last updated (optional).
Comment	Describes comments about the description to which the description metadata is attached using a textual annotation (optional).
PublicIdentifier	Identifies the description to which the description metadata is attached using a public, globally unique identifier (optional). This identifier may be used to uniquely identify descriptions and is not dependent on the file/stream in which the description occurs, or any other physical embodiment of the metadata. Multiple public identifiers may be associated with a description.
PrivateIdentifier	Identifies the description to which the description metadata is attached using a private, application dependent identifier (optional). The format of this identifier is application defined and need not be unique. Multiple private identifiers may be associated with a description.
Creator	Describes a creator of the description to which the description metadata is attached (optional). This can be a person, organization, or the software application that automatically generated the metadata. Multiple creators are allowed if the metadata was created as the result of several creators cooperating. Creator is of type CreatorType, which is defined in 9.2.
CreationLocation	Describes the place where the description to which the description metadata is attached was created (optional). CreationLocation is of type PlaceType, which is defined in 7.6.
CreationTime	Describes the time when the description to which the description metadata is attached was created (optional).
Instrument	Describes the devices/procedure and settings used for the creation of the metadata, such as the tools used to extract the metadata or the extraction parameters (optional). Instrument is of type CreationToolType, which is defined in 9.2.
Rights	Describes the rights associated with the description to which the description metadata is attached and how the description to which this DS is attached can be used. (optional). Rights is of type RightsType, which is defined in 10.2.2.

Name	Definition
Package	Describes a package associated with the description that describes the tools that are used by the description (optional). Package is of type PackageType, which is defined in 4.5.2.

4.5.3.4 DescriptionMetadata header examples (informative)

The following example shows the use of the DescriptionMetadata Header for describing a video segment.

Note: 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">
          <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>
            </Header>
          </VideoSegment>
        </TemporalDecomposition>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```

        </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="SemanticType"/>
        <Scheme name="UserPreferencesType"/>
    </Package>
    </Header>
    <!-- more elements here -->
</VideoSegment>                                <!-- more elements here -->
</TemporalDecomposition>
<!-- more elements here -->
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

5 Basic datatypes

5.1 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 5 — Overview of Basic Datatypes

<i>Tool</i>	<i>Functionality</i>
Integer & Real datatypes	Describes 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: minusOneToOne, zeroToOne, and so on.
Vectors & Matrix datatypes	Describes arbitrary sized vectors and matrices of integer or real values. For vectors, the IntegerVector, FloatVector, and DoubleVector datatypes represent vectors of integer, float, and double values respectively. For matrices, the IntegerMatrix, FloatMatrix, and DoubleMatrix datatypes represent matrices of integer, float, and double values respectively.
Probability Vectors & Matrix datatypes	Describes probability distributions using vectors (ProbabilityVector) and matrices (ProbabilityMatrix).
String Datatypes	Describes content types (mimeType), countries (countryCode), regions (regionCode), currencies (currencyCode), and character sets (characterSetCode).

5.2 Integer datatypes

5.2.1 Introduction

This subclause defines datatypes that represent different kinds of constrained integer types.

5.2.2 Unsigned datatypes

5.2.2.1 Introduction

The "unsignedXX" datatypes represent integer values constrained to a fixed number ("xx") of bits. There is one datatype for each bit length, from one bit (unsigned1) to thirty-two bits (unsigned32).

5.2.2.2 Unsigned datatypes syntax

```
<!-- ##### -->
<!-- Definition of Unsigned datatypes (5.2.2) -->
<!-- ##### -->

<!-- Definition of unsigned1 datatype -->
<simpleType name="unsigned1">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="1"/>
  </restriction>
</simpleType>
```

```

    </restriction>
  </simpleType>

  <!-- Definition of unsigned2 datatype -->
  <simpleType name="unsigned2">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="3"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned3 datatype -->
  <simpleType name="unsigned3">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="7"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned4 datatype -->
  <simpleType name="unsigned4">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="15"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned5 datatype -->
  <simpleType name="unsigned5">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="31"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned6 datatype -->
  <simpleType name="unsigned6">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="63"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned7 datatype -->
  <simpleType name="unsigned7">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="127"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned8 datatype -->
  <simpleType name="unsigned8">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="255"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned9 datatype -->

```

```

<simpleType name="unsigned9">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="511"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned10 datatype -->
<simpleType name="unsigned10">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="1023"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned11 datatype -->
<simpleType name="unsigned11">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="2047"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned12 datatype -->
<simpleType name="unsigned12">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="4095"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned13 datatype -->
<simpleType name="unsigned13">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="8191"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned14 datatype -->
<simpleType name="unsigned14">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="16383"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned15 datatype -->
<simpleType name="unsigned15">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="32767"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned16 datatype -->
<simpleType name="unsigned16">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="65535"/>
  </restriction>
</simpleType>

```



```

    </restriction>
  </simpleType>

  <!-- Definition of unsigned17 datatype -->
  <simpleType name="unsigned17">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="131071"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned18 datatype -->
  <simpleType name="unsigned18">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="262143"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned19 datatype -->
  <simpleType name="unsigned19">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="524287"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned20 datatype -->
  <simpleType name="unsigned20">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="1048575"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned21 datatype -->
  <simpleType name="unsigned21">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="2097151"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned22 datatype -->
  <simpleType name="unsigned22">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="4194303"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned23 datatype -->
  <simpleType name="unsigned23">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="8388607"/>
    </restriction>
  </simpleType>

  <!-- Definition of unsigned24 datatype -->

```

```

<simpleType name="unsigned24">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="16777215"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned25 datatype -->
<simpleType name="unsigned25">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="33554431"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned26 datatype -->
<simpleType name="unsigned26">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="67108863"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned27 datatype -->
<simpleType name="unsigned27">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="134217727"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned28 datatype -->
<simpleType name="unsigned28">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="268435455"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned29 datatype -->
<simpleType name="unsigned29">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="536870911"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned30 datatype -->
<simpleType name="unsigned30">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="1073741823"/>
  </restriction>
</simpleType>

<!-- Definition of unsigned31 datatype -->
<simpleType name="unsigned31">
  <restriction base="nonNegativeInteger">
    <minInclusive value="0"/>
    <maxInclusive value="2147483647"/>
  </restriction>
</simpleType>

```

```

    </restriction>
  </simpleType>

  <!-- Definition of unsigned32 datatype -->
  <simpleType name="unsigned32">
    <restriction base="nonNegativeInteger">
      <minInclusive value="0"/>
      <maxInclusive value="4294967295"/>
    </restriction>
  </simpleType>

```

5.2.2.3 Unsigned datatypes semantics

Semantics of the unsigned datatypes:

<i>Name</i>	<i>Definition</i>
unsigned1	A 1-bit unsigned integer value.
unsigned2	A 2-bit unsigned integer value.
unsigned3	A 3-bit unsigned integer value.
unsigned4	A 4-bit unsigned integer value.
unsigned5	A 5-bit unsigned integer value.
unsigned6	A 6-bit unsigned integer value.
unsigned7	A 7-bit unsigned integer value.
unsigned8	A 8-bit unsigned integer value.
unsigned9	A 9-bit unsigned integer value.
unsigned10	A 10-bit unsigned integer value.
unsigned11	A 11-bit unsigned integer value.
unsigned12	A 12-bit unsigned integer value.
unsigned13	A 13-bit unsigned integer value.
unsigned14	A 14-bit unsigned integer value.
unsigned15	A 15-bit unsigned integer value.
unsigned16	A 16-bit unsigned integer value.
unsigned17	A 17-bit unsigned integer value.
unsigned18	A 18-bit unsigned integer value.
unsigned19	A 19-bit unsigned integer value.
unsigned20	A 20-bit unsigned integer value.

Name	Definition
unsigned21	A 21-bit unsigned integer value.
unsigned22	A 22-bit unsigned integer value.
unsigned23	A 23-bit unsigned integer value.
unsigned24	A 24-bit unsigned integer value.
unsigned25	A 25-bit unsigned integer value.
unsigned26	A 26-bit unsigned integer value.
unsigned27	A 27-bit unsigned integer value.
unsigned28	A 28-bit unsigned integer value.
unsigned29	A 29-bit unsigned integer value.
unsigned30	A 30-bit unsigned integer value.
unsigned31	A 31-bit unsigned integer value.
unsigned32	A 32-bit unsigned integer value.

5.3 Real datatypes

5.3.1 Introduction

This subclause defines datatypes that represent real values constrained to lie within a specified range. In this subclause, *real value* is used to refer to the set of real numbers represented as floating point values, without specifying the precision of that representation. The DDL provides two types of real values: float and double (see ISO/IEC 15938-2).

5.3.2 zeroToOne datatype

5.3.2.1 Introduction

The zeroToOne datatype represents a float value in the range from 0.0 to 1.0.

5.3.2.2 zeroToOne datatype syntax

```
<!-- ##### -->
<!-- Definition of zeroToOne datatype (5.3.2) -->
<!-- ##### -->

<!-- Definition of zeroToOne datatype -->
<simpleType name="zeroToOneType">
  <restriction base="float">
    <minInclusive value="0.0"/>
    <maxInclusive value="1.0"/>
  </restriction>
</simpleType>
```

5.3.2.3 zeroToOne datatype semantics

Semantics of the zeroToOneType:

Name	Definition
zeroToOneType	A float value in the range 0.0 to 1.0 inclusive.

5.3.3 minusOneToOne datatype

5.3.3.1 Introduction

The minusOneToOne datatype represents a float value in the range from -1.0 to 1.0.

5.3.3.2 minusOneToOne datatype syntax

```
<!-- ##### -->
<!-- Definition of minusOneToOne datatype (5.3.3) -->
<!-- ##### -->

<!-- Definition of minusOneToOne datatype -->
<simpleType name="minusOneToOneType">
  <restriction base="float">
    <minInclusive value="-1.0"/>
    <maxInclusive value="+1.0"/>
  </restriction>
</simpleType>
```

5.3.3.3 minusOneToOne datatype semantics

Semantics of the minusOneToOneType:

Name	Definition
minusOneToOneType	A float value in the range -1.0 to 1.0 inclusive.

5.3.4 nonNegativeReal datatype

5.3.4.1 Introduction

The nonNegativeReal datatype represents a double value in the range $[0, \infty)$.

5.3.4.2 nonNegativeReal datatype syntax

```
<!-- ##### -->
<!-- Definition of nonNegativeReal datatype (5.3.4) -->
<!-- ##### -->

<!-- Definition of nonNegativeReal datatype -->
```

```
<simpleType name="nonNegativeReal">
  <restriction base="double">
    <minInclusive value="0.0"/>
  </restriction>
</simpleType>
```

5.3.4.3 nonNegativeReal datatype semantics

Semantics of the nonNegativeReal:

Name	Definition
nonNegativeReal	A double value in the range 0.0 (inclusive) to positive infinity.

5.3.5 Real datatypes examples (informative)

The following example shows the use of the zeroToOne integer datatype for describing the confidence of a model:

```
<Model confidence="1.0">
  <!-- more elements here -->
</Model>
```

5.4 Vectors and matrices

5.4.1 Introduction

This subclause defines datatypes that represent arbitrary sized vectors and matrices of either integer or real numbers. Matrices and vectors are arrangements of values into tables where each value is assigned an index and are commonly used as datatypes for representing discrete signals—for example, an audio signal can be represented as a 1-dimensional vector or a video as a 3-dimensional matrix—and to represent features of multimedia content, such as a color histogram.

This subclause uses the following terminology. A **matrix** is arrangement of items along a set of **n** finite axes, called the **dimensions** of the matrix. Formally, a matrix *M* is a mapping from $N_{d1} \times N_{d2} \times \dots \times N_{dn}$ to a set of items, which depend on the type of the matrix. N_j represents the set of natural numbers $\{1, \dots, j\}$, d_k is the **size** of the k^{th} dimension of the matrix *M* and *n* is the number of dimensions in the matrix. A matrix is **n-dimensional** if it has *n* dimensions (it is not limited to being two-dimensional as it common in linear algebra). The first dimension is called dimension one, the second dimension two, and so one. A **vector** is a 1-dimensional matrix.

5.4.1.1 Representing vector and matrix values (Informative)

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.

5.4.2 vector datatypes

5.4.2.1 Introduction

The `integerVector`, `floatVector`, and `doubleVector` datatypes represent ordered sequences of integer, float, and double values, respectively.

5.4.2.2 vector datatypes syntax

```
<!-- ##### -->
<!-- Definition of vector datatypes (5.4.2) -->
<!-- ##### -->

<!-- Definition of integerVector datatype -->
<simpleType name="integerVector">
  <list itemType="integer"/>
</simpleType>

<!-- Definition of floatVector datatype -->
<simpleType name="floatVector">
  <list itemType="float"/>
</simpleType>

<!-- Definition of doubleVector datatype -->
<simpleType name="doubleVector">
  <list itemType="double"/>
</simpleType>
```

5.4.2.3 vector datatypes semantics

Semantics of the `integerVector`:

Name	Definition
<code>integerVector</code>	A one-dimensional vector of integer values. The size of the vector is determined by the number of integer values present in the vector's instance.

Semantics of the floatVector:

Name	Definition
floatVector	A one-dimensional vector of float values. The size of the vector is determined by the number of float values present in the vector's instance.

Semantics of the doubleVector:

Name	Definition
doubleVector	A one-dimensional vector of double values. The size of the vector is determined by the number of double values present in the vector's instance.

Formally, a vector is defined as a mapping from the finite integer subset $\{1, \dots, n\}$ to the set of integer, float, or double values (according to the type of the vector), where n is the size of the vector, or equivalently, the number of values occurring in the vector instance. Let $V(i)$, $1 \leq i \leq n$ represent the i^{th} element of the vector. Then, the value of a vector is specified by exhaustively enumerating the values of the vector (mapping) in the order $V(1), V(2), \dots, V(n)$.

5.4.2.4 vector datatypes examples (informative)

The following example shows the use of the floatVector datatype for describing the MotionParam of a mosaic:

```
<Mosaic>
  <WarpingParam modelType="affine">
    <!-- more elements here -->
    <MotionParam> -3.60 0.73 0.03 0.01 0.01 0.51 </MotionParam>
    <!-- more elements here -->
  </WarpingParam>
</Mosaic>
```

5.4.3 minusOneToOneVector datatype

5.4.3.1 Introduction

The minusOneToOneVector datatype represents a vector of float values constrained to lie in the range -1.0 to 1.0 , inclusive.

5.4.3.2 minusOneToOneVector datatype syntax

```
<!-- ##### -->
<!-- Definition of minusOneToOneVector datatype (5.4.3) -->
<!-- ##### -->

<!-- Definition of minusOneToOneVector datatype -->
<simpleType name="minusOneToOneVector">
  <list>
    <simpleType>
      <restriction base="float">
        <minInclusive value="-1.0"/>
        <maxInclusive value="1.0"/>
      </restriction>
    </simpleType>
  </list>
</simpleType>
```


5.4.3.3 minusOneToOneVector datatype semantics

Semantics of the minusOneToOneVector:

Name	Definition
minusOneToOneVector	A vector of float values constrained to the range [-1.0,1.0].

5.4.4 Matrix datatypes

5.4.4.1 Introduction

The IntegerMatrix, FloatMatrix, and DoubleMatrix datatypes represent n -dimensional arrays of integer, float, and double values, respectively, where n is greater than or equal to one.

5.4.4.2 Matrix datatypes syntax

```

<!-- ##### -->
<!-- Definition of Matrix datatypes (5.4.4) -->
<!-- ##### -->

<!-- Definition of IntegerMatrix datatype -->
<complexType name="IntegerMatrixType">
  <simpleContent>
    <extension base="mpeg7:integerVector">
      <attribute ref="mpeg7:dim" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!-- Definition of FloatMatrix datatype -->
<complexType name="FloatMatrixType">
  <simpleContent>
    <extension base="mpeg7:floatVector">
      <attribute ref="mpeg7:dim" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!-- Definition of DoubleMatrix datatype -->
<complexType name="DoubleMatrixType">
  <simpleContent>
    <extension base="mpeg7:doubleVector">
      <attribute ref="mpeg7:dim" use="required"/>
    </extension>
  </simpleContent>
</complexType>

```

5.4.4.3 Matrix datatypes semantics

Semantics of the IntegerMatrixType:

Name	Definition
IntegerMatrixType	An n-dimensional matrix of integer values.
dim	Indicates the dimensions of the matrix. The number of elements in the list determines the number of dimensions in the matrix. Each value in the list determines the size of the corresponding dimension—the first value is the size of the first dimension, the second value the size of the second dimension, and so on—and shall be greater than zero.

Semantics of the FloatMatrixType:

Name	Definition
FloatMatrixType	An n-dimensional matrix of float values.
dim	Indicates the dimensions of the matrix. The number of elements in the list determines the number of dimensions in the matrix. Each value in the list determines the size of the corresponding dimension—the first value is the size of the first dimension, the second value the size of the second dimension, and so on—and shall be greater than zero.

Semantics of the DoubleMatrixType:

Name	Definition
DoubleMatrixType	An n-dimensional matrix of double values.
dim	Indicates the dimensions of the matrix. The number of elements in the list determines the number of dimensions in the matrix. Each value in the list determines the size of the corresponding dimension—the first value is the size of the first dimension, the second value the size of the second dimension, and so on—and shall be greater than zero.

Formally a matrix M is a mapping from $N_{d1} \times N_{d2} \times \dots \times N_{dn}$ to the set of integers, floats, or doubles (according to the type of the vector), where N_j represents the set of natural numbers $\{1, \dots, j\}$, d_k is the size of the k^{th} dimension of the matrix and n is the dimension of the matrix. The values of d_k are given by the dim attribute in order of occurrence in the list. The length of this list is the value of n . The value of a matrix is specified by exhaustively enumerating its elements in the following pre-defined order

$(1, 1, \dots, 1)$
 $(1, 1, \dots, 2)$
 $\dots$
 $(1, 1, \dots, d_n)$
 $(1, 1, \dots, 2, 1)$
 $\dots$
 $(1, 1, d_{n-1}, d_n)$
 $\dots$
 $(d_1, d_2, \dots, d_n)$

In other words, the last subscript varies fastest, the second to last subscript varies second fastest, and so on. For 2-D dimensional matrices, this corresponds to what is commonly called "row-major" order. For example, for a $k \times l$ 2-dimensional matrix this means that the elements would appear in the order: (1,1), (1,2) (1, l), (2, 1), (2,2),....., (2, l), ... (k, 1), (k, 2), ..., (k,l).

5.4.4.4 Matrix datatypes examples (informative)

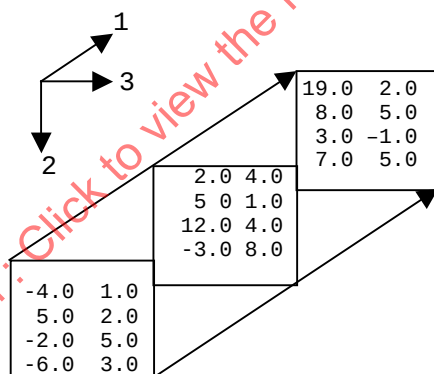
Consider the two-dimensional 4x4 matrix shown as follows:

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

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>
```

5.4.5 Diagonal matrix datatypes

5.4.5.1 Introduction

The IntegerDiagonalMatrix, FloatDiagonalMatrix, and DoubleDiagonalMatrix datatypes represent a diagonal matrix (a square two-dimensional matrix whose non-diagonal elements are all zero) with a vector containing only the diagonal components.

5.4.5.2 Diagonal matrix datatypes syntax

```
<!-- ##### -->
<!-- Definition of DiagonalMatrix datatypes (5.4.5) -->
<!-- ##### -->

<!-- Definition of IntegerDiagonalMatrix datatype -->
<complexType name="IntegerDiagonalMatrixType">
  <simpleContent>
    <extension base="mpeg7:integerVector">
      <attribute ref="mpeg7:dim" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!-- Definition of FloatDiagonalMatrix datatype -->
<complexType name="FloatDiagonalMatrixType">
  <simpleContent>
    <extension base="mpeg7:floatVector">
      <attribute ref="mpeg7:dim" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!-- Definition of DoubleDiagonalMatrix datatype -->
<complexType name="DoubleDiagonalMatrixType">
  <simpleContent>
    <extension base="mpeg7:doubleVector">
      <attribute ref="mpeg7:dim" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

5.4.5.3 Diagonal matrix datatypes semantics

Semantics of the IntegerDiagonalMatrixType:

Name	Definition
IntegerDiagonalMatrixType	A square 2-dimensional diagonal matrix represented by a vector (1-dimensional matrix) containing only the diagonal elements of the matrix, whose values have integer datatype.
dim	Indicates the size of the diagonal vector of the square matrix. If this value is n , then the resulting matrix is n -by- n .

Semantics of the FloatDiagonalMatrixType:

Name	Definition
FloatDiagonalMatrixType	A square 2-dimensional diagonal matrix represented by a vector (1-dimensional matrix) containing only the diagonal elements of the matrix, whose values have float datatype.
dim	Indicates the length of the diagonal vector of the square matrix. If this value is n , then the resulting matrix is n -by- n .

Semantics of the DoubleDiagonalMatrixType:

Name	Definition
DoubleDiagonalMatrixType	A square 2-dimensional diagonal matrix represented by a vector (1-dimensional matrix) containing only the diagonal elements of the matrix, whose values have double datatype.
dim	Indicates the length of the diagonal vector of the square matrix. If this value is n , then the resulting matrix is n -by- n .

Formally, a diagonal matrix is defined as follows: denote the element at index (i,j) , $1 \leq i \leq n$, $1 \leq j \leq n$, of the diagonal matrix M (of size $n \times n$) by $M(i,j)$. Let V be the value of the diagonal vector and $V(i)$ the diagonal vector element at index i , $1 \leq i \leq n$. Then the value of the element $M(i,j)$ of the full diagonal matrix is defined as follows:

$$M(i,j) = \begin{cases} V(i) & i = j \\ 0 & i \neq j \end{cases}$$

5.4.5.4 Diagonal matrix datatypes examples (informative)

Consider the diagonal matrix shown as follows:

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

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>
```

5.4.6 MinusOneToOneMatrix datatype

5.4.6.1 Introduction

The MinusOneToOneMatrix datatype represents a matrix whose values are constrained to lie in the range from minus one to one.

5.4.6.2 MinusOneToOneMatrix datatype syntax

```
<!-- ##### -->
<!-- Definition of MinusOneToOneMatrix datatype (5.4.6) -->
<!-- ##### -->

<!-- Definition of MinusOneToOneMatrix datatype -->
<complexType name="MinusOneToOneMatrixType">
  <simpleContent>
    <extension base="mpeg7:minusOneToOneVector">
      <attribute ref="mpeg7:dim" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

5.4.6.3 MinusOneToOneMatrix datatype semantics

Semantics of the MinusOneToOneMatrixType:

Name	Definition
MinusOneToOneMatrixType	An n-dimensional matrix of values where each value in the matrix shall be in the range from minus one to one.
dim	Indicates the size of the matrix with a list of integer values. The number of elements in the list determines the dimension of the matrix. Each element in the list, which shall be greater than zero, determines the size of one dimension of the matrix.

5.5 Probability datatypes

5.5.1 Introduction

The probability datatypes specified in this subclause represent unconditional and conditional discrete probability distributions using vectors and matrices.

5.5.2 probabilityVector datatype

5.5.2.1 Introduction

The probabilityVector datatype represents a probability distribution over a finite domain (the set of possible outcomes) as a vector. The value of each vector component defines the probability value for the corresponding element in the finite domain.

Note: The tool that uses this datatype defines the domain of the probability distribution.

5.5.2.2 probabilityVector datatype syntax

```

<!-- ##### -->
<!-- Definition of probabilityVector datatype (5.5.2) -->
<!-- ##### -->

<!-- Definition of probabilityVector datatype -->
<simpleType name="probabilityVector">
  <list>
    <simpleType>
      <restriction base="float">
        <minInclusive value="0.0"/>
        <maxInclusive value="1.0"/>
      </restriction>
    </simpleType>
  </list>
</simpleType>

```

5.5.2.3 probabilityVector datatype semantics

Semantics of the probabilityVector:

Name	Definition
probabilityVector	A vector of probability values. Each value shall be in the range from 0.0 to 1.0 and the values in the vector shall sum to one.

The correspondence between the set of possible outcomes and the vector components shall be defined in the tool that uses this datatype.

5.5.3 ProbabilityMatrix datatype

5.5.3.1 Introduction

The ProbabilityMatrix datatype represents a joint or conditional probability table. Each finite random variable in the conditional or joint distribution is represented by one dimension (axis) in the matrix. The size of each dimension is the number of possible values for the corresponding random variable. For example, a conditional distribution $P(W|S,R)$ involving three random variables (W , S and R) would be represented as a 3-dimensional matrix. Similarly, a joint distribution $P(S, R, W)$ would also be represented by a 3-dimensional matrix.

This datatype does not specify the correspondence between random variables in a distribution and dimensions of the ProbabilityMatrix. The context of use (i.e. datatype, descriptor, or description scheme using this datatype) shall specify the correspondence between the variables of the distribution (i.e. the S , R , and W in a distribution like $P(R, S, W)$) correspond to each matrix dimension in the ProbabilityMatrix. Furthermore, the context of use shall specify whether the distribution being represented is a conditional or joint distribution.

5.5.3.2 ProbabilityMatrix datatype syntax

```

<!-- ##### -->
<!-- Definition of ProbabilityMatrix datatype (5.5.3) -->
<!-- ##### -->

<!-- Definition of probabilityMatrix datatype -->
<complexType name="ProbabilityMatrixType">
  <simpleContent>

```

```

<extension base="mpeg7:probabilityVector">
  <attribute ref="mpeg7:dim" use="required"/>
</extension>
</simpleContent>
</complexType>

```

5.5.3.3 ProbabilityMatrix datatype semantics

Semantics of the ProbabilityMatrixType:

Name	Definition
ProbabilityMatrixType	An n-dimensional matrix representing a joint or conditional probability distribution involving n different random variables. Let $P(X_{i+1}, \dots, X_n X_1, \dots, X_i)$ be a conditional probability (joint if i is zero) distribution where $X_1, \dots, X_n$ are the random variables. Then elements at index $(a_1, \dots, a_n)$ of the probability matrix represents the probability $P(X_{i+1}=a_{i+1}, \dots, X_n=a_n X_1=a_1, \dots, X_i=a_i)$. Each value in the matrix shall be in the range from zero to one.
dim	Indicates the dimensions of the probability matrix with a list of integer values. The number of elements in the list determines the number of dimension in the matrix. Each element in the list determines the size of the corresponding dimension in the matrix and shall greater than zero. Each dimension of the matrix represents one random variable and the size of that dimension is equal to number of values possible for that variable.

5.5.3.4 ProbabilityMatrix datatype examples (informative)

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 6 — Example of conditional probabilities

R	S	$P(W=false R,S)$	$P(W=true R,S)$
false	false	1.00	0.00
false	true	0.10	0.90
true	false	0.80	0.20
true	true	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: other assignments of variables to indices can be used.

5.6 String datatypes

5.6.1 Introduction

The datatypes defined in this subclause define various kinds of coded string representations as follows.

- mimeType - IANA list of MIME content types (MIMETYPE).
- countryCode -- Country codes as specified in ISO 3166-1:1997 (ISO 3166-1).
- regionCode -- Region codes as specified in ISO 3166-2:1998 (ISO 3166-2).
- currencyCode -- Currency codes as specified in ISO 4217:1995 (ISO 4217).
- characterSetCode - Character set codes as defined in IANA List of Character Sets (RFC2045-CHARSETS).

5.6.2 mimeType datatype

5.6.2.1 Introduction

The mimeType datatype describes a content type, i.e. media type. A current list of registered content types is available from the IANA List of Mime Types (RFC2045-MIMETYPES). Examples of content types include "text/xml," "image/jpeg," and "video/mpeg."

5.6.2.2 mimeType datatype syntax

```
<simpleType name="mimeType">
  <restriction base="string">
    <whiteSpace value="collapse"/>
    <pattern value='[&#x21;-&#x7f;-[\(\)&lt;&gt;@,;:\\"/\[\]\?]=])+/[&#x21;-&#x7f;-[\(\)&lt;&gt;@,;:\\"/\[\]\?]=]+'/'>
  </restriction>
</simpleType>
```

5.6.2.3 mimeType datatype semantics

Semantics of the mimeType:

Name	Definition
contentType	Identifies the content type (i.e. media type) of a resource using a value adhering to the syntax and semantics defined in RFC2045. The value is case-insensitive.
mediaType	Identifies the media type (i.e. media type) of a resource using a value adhering to the syntax and semantics defined in RFC2045. The value is case-insensitive.
resourceType	Identifies the resource type (i.e. media type) of a resource using a value adhering to the syntax and semantics defined in RFC2045. The value is case-insensitive.
contentType	Identifies the content type (i.e. media type) of a resource using a value adhering to the syntax and semantics defined in RFC2045. The value is case-insensitive.
mediaType	Identifies the media type (i.e. media type) of a resource using a value adhering to the syntax and semantics defined in RFC2045. The value is case-insensitive.
resourceType	Identifies the resource type (i.e. media type) of a resource using a value adhering to the syntax and semantics defined in RFC2045. The value is case-insensitive.

5.6.3 countryCode datatype

5.6.3.1 Introduction

The countryCode datatype identifies a country using the two letter country codes (ISO 3166-1). The current list of valid country codes is available from the official maintenance authority.

5.6.3.2 countryCode datatype syntax

```
<!-- ##### -->
<!-- Definition of countryCode datatype (5.6.3) -->
<!-- ##### -->

<!-- Definition of countryCode datatype -->
<simpleType name="countryCode">
  <restriction base="string">
    <pattern value="[a-zA-Z]{2}" />
    <whiteSpace value="collapse" />
  </restriction>
</simpleType>
```

5.6.3.3 countryCode datatype semantics

Semantics of the countryCode:

Name	Definition
countryCode	Identifies a country using a valid ISO 3166-1 Alpha-2 code, which is a two letter code identifying a country defined in ISO 3166-1. The value is case-insensitive.

5.6.3.4 countryCode datatype example (informative)

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

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

5.6.4 regionCode datatype

5.6.4.1 Introduction

The regionCode datatype identifies a region within a country using the region codes defined in ISO 3166-2:1998 (ISO 3166-2). The current list of valid region codes is available from the official maintenance authority.

Note: ISO 3166-2 provides a breakdown into relevant administrative subdivisions of all countries listed in ISO 3166-1. A region code consists of up to three alphanumeric characters, e.g. "MI" for the Italian province of Milan. The characters encoding a region code are unique only within a particular country. There are overlaps of region codes for different countries.

5.6.4.2 regionCode datatype syntax

```
<!-- ##### -->
<!-- Definition of regionCode datatype (5.6.4) -->
<!-- ##### -->

<!-- Definition of regionCode datatype -->
<simpleType name="regionCode">
  <restriction base="string">
    <pattern value="[a-zA-Z]{2}(-[a-zA-Z0-9]{1,3})?" />
    <whiteSpace value="collapse" />
  </restriction>
</simpleType>
```

5.6.4.3 regionCode datatype semantics

Semantics of the regionCode:

Name	Definition
regionCode	Describes a region within a country using the administrative subdivision (region) codes defined in ISO 3166-1. A region code consists of the 2-letter country code, optionally followed by a dash ("-") separator and the code for the administrative subdivision within that country, which consists of up to three alphanumeric characters. For example, "IT-MI" designates the Italian province of Milan. When the region code is omitted, the entire country is being described. Note: the characters encoding a region code are unique only within a particular country. There are overlaps of region codes for different countries. Note: region codes are case insensitive

5.6.4.4 regionCode datatype example (informative)

The following examples show the use of regionCode for describing places (the Italian province of Milano and Danish county Roskilde) using region codes:

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

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

The following example shows the use of mixed case to illustrate case-insensitivity of the country/region codes:

```
<Place>
  <Name>Antananarivo province in Madagascar</Name>
  <Region>mG-t</Region>
</Place>
```

5.6.5 currencyCode datatype

5.6.5.1 Introduction

The currencyCode datatype identifies a currency using the codes defined in ISO 4217. The current list of currency codes is available from the official maintenance authority.

5.6.5.2 currencyCode datatype syntax

```
<!-- ##### -->
<!-- Definition of currencyCode datatype (5.6.5) -->
<!-- ##### -->

<!-- Definition of currencyCode datatype -->
<simpleType name="currencyCode">
  <restriction base="string">
    <pattern value="[a-zA-Z]{3}" />
    <whiteSpace value="collapse" />
  </restriction>
</simpleType>
```

5.6.5.3 currencyCode datatype semantics

Semantics of the currencyCode:

Name	Definition
currencyCode	Identifies a currency or fund using a three-level alphabetic code defined in ISO 4217. A currency code is a three-letter alphabetic code. The value is case-insensitive. The currency code is usually composed of the country's two-character ISO 3166-1 country code, plus a one-character currency designator. For example, the code for Australian Dollars is Australia's two-character ISO 3166-1 code (AU) plus a one-character currency designator (D).

5.6.5.4 currencyCode datatype example (informative)

The following shows several examples of different currency codes:

Currency Name	Currency Code
Hong Kong Dollar	HKD
Euro	EUR

5.6.6 characterSetCode datatype

5.6.6.1 Introduction

The characterSetCode datatype identifies a character set using a code conforming to RFC2045-CHARSETS. The current list of registered character set code is maintained in the IANA list of character sets.

5.6.6.2 characterSetCode datatype syntax

```

<!-- ##### -->
<!-- Definition of characterSetCode datatype (5.6.6) -->
<!-- ##### -->

<!-- Definition of characterSetCode datatype -->
<simpleType name="characterSetCode">
  <restriction base="string"/>
</simpleType>

```

5.6.6.3 characterSetCode datatype semantics

Semantics of the characterSetCode:

Name	Definition
characterSetCode	Character set code identifying a character set using the set of character set codes maintained by the IANA. Character set names may be up to 40 characters taken from the printable characters of US-ASCII. The value is case-insensitive. The value is not restricted to being one of the registered IANA character set codes as long as it conforms to the requirements for character set codes set out in RFC2045-CHARSETS.

5.6.6.4 characterSetCode datatype example (informative)

The following table shows several examples of character set codes:

Table 7 — 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

6 Linking, identification and localization tools

6.1 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 8 — Overview of Linking, Identification, and Localization Tools

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

6.2 References to Ds and DSs

6.2.1 Reference datatype

6.2.1.1 Introduction

The Reference datatype is used by description tools to refer to a part of the description (i.e., an instance of a D or DS). The Reference datatype allows three types of referencing mechanisms:

- 1) AnyURI: which references a description that is outside the document which contains the current description,
- 2) IDREF: which references elements that have an instantiated attribute of type ID. The referenced elements are within the document containing the current description,
- 3) XPath expression: which references an element or attribute within the document containing the current description (The XPath expression defined by the XPathRefType is a subset of the XPath definition in (XPath). It is limited with respect to the axis and the predicates. For a detailed specification see Annex A).

The IDREF allows the existence of the target to be verified. The XPath mechanism allows the reference target to have an attribute of type other than ID. Only one of the mechanisms can be used at a time. These referencing mechanisms are used within Ds or DSs by referring to the referenceGrp attribute group defined in this clause or by defining an element of type ReferenceType.

6.2.1.2 Reference datatype syntax

```

<!-- ##### -->
<!-- Definition of Reference datatypes (6.2.1) -->
<!-- ##### -->

<!-- Definition of referenceGrp attribute group -->
<attributeGroup name="referenceGrp">
  <attribute name="idref" type="IDREF" use="optional"/>
  <attribute name="xpath" type="mpeg7:xPathRefType" use="optional"/>
  <attribute name="href" type="anyURI" use="optional"/>
</attributeGroup>

<!-- Definition of Reference datatype -->
<complexType name="ReferenceType">
  <attributeGroup ref="mpeg7:referenceGrp"/>
</complexType>

```

6.2.1.3 Reference datatype semantics

Semantics of the ReferenceType:

Name	Definition
referenceGrp	Describes a reference using one of the following mechanisms: idref, xpath and href. The semantics of the reference is specified within the definition of the D or DS that refers to this attribute group. The idref, xpath and href referencing mechanisms are alternative mechanisms which can be used to describe a reference. Only one these reference mechanisms shall be used at a time in an instance of referenceGrp.
idref	Identifies a referenced element that is contained within the same description that contains the reference (optional). The value of this attribute, when used, shall match the value of the attribute of type ID of the referenced element.
xpath	Identifies a referenced element or attribute that is contained in the current description using XPath specification (optional).
href	Identifies a referenced element that is not contained within the current description. The href describes the reference to the description which contains the target element and the ID of this element (optional).
ReferenceType	Describes a reference to an element of a description. The reference is described by instantiating one of the attributes contained in the referenceGrp attribute group.

6.2.1.4 Reference datatype examples (informative)

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 11 and clause 12). 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 of type ID "id3" 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 "another-description.mp7".

```
<MyElement href="www.example.com/another-description.mp7#id1">...</MyElement>
```

6.2.2 XPath datatypes

6.2.2.1 Introduction

XPath expressions are used to identify elements and attributes in the current description. Based on a generic XPath datatype three specific datatypes are defined:

- 1) `xPathRefType` describes a single element or attribute within the current description. This datatype is used for instance in the reference datatype to point to a single target element or attribute.
- 2) `xPathSelectorType` describes a set of elements within the current description for instance to select a set of DSs according to the specification in the `OrderingKey` DS (see 7.8).
- 3) `xPathAbsoluteSelectorType` describes a set of elements within the current description using only absolute XPath specifications for instance to select a set of DSs according to the specification in the `ClassificationScheme` DS (see 7.4.2).
- 4) `xPathFieldType` describes a set of elements or attributes within the current description for instance to identify the fields on which an ordering of elements is based on (see 7.8).

The XPath expressions specified in these datatypes are subsets of the XPath definition in (XPath). They are limited with respect to the axis and the predicates. For a detailed specification see Annex A.

6.2.2.2 XPath datatypes syntax

```
<!-- ##### -->
<!-- Definition of XPath datatypes (6.2.2) -->
<!-- ##### -->

<!-- Definition of XPath datatype -->
<simpleType name="XPathType">
  <restriction base="token"/>
</simpleType>

<!-- Definition of XPathRef datatype -->
<simpleType name="XPathRefType">
  <restriction base="mpeg7:XPathType">
    <pattern
value="/?(((child:)?((\i\c*:)?(\i\c*)(\[\d+\]))?)|\.|(\.\.))/)*(((child:)?((\i\c*:)?(\i\c*)(\[\d+\]))?)|\.|((attribute:|@)((\i\c*:)?(\i\c*|*))))"/>
    </restriction>
  </simpleType>

<!-- Definition of XPathSelector datatype -->
<simpleType name="XPathSelectorType">
  <restriction base="mpeg7:XPathType">
    <pattern
value="(\/|((//|/)?(((child:)?((\i\c*:)?(\i\c*|*)))|\.|(\.\.))((//|/)?(((child:)?((\i\c*:)?(\i\c*|*)))|\.|(\.\.)))*))(\|/|((//|/)?(((child:)?((\i\c*:)?(\i\c*|*)))|\.|(\.\.))((//|/)?(((child:)?((\i\c*:)?(\i\c*|*)))|\.|(\.\.)))*))*"/>
    </restriction>
  </simpleType>

<!-- Definition of XPathAbsoluteSelector datatype -->
<simpleType name="XPathAbsoluteSelectorType">
  <restriction base="mpeg7:XPathSelectorType">
    <pattern
value="(\/|((//|/)?(((child:)?((\i\c*:)?(\i\c*|*)))|\.|(\.\.)))*)(\|/|((//|/)?(((child:)?((\i\c*:)?(\i\c*|*)))|\.|(\.\.)))*))*"/>
```



```

    </restriction>
  </simpleType>

  <!-- Definition of XPathField datatype -->
  <simpleType name="XPathFieldType">
    <restriction base="mpeg7:XPathType">
      <pattern
value="(\.//)?((((child:)?((\i\c*:)?(\i\c*|\*)))|\.)/)*((((child:)?((\i\c*:)?(\i\c*|\*)))|\.)|((attribute::|@)((\i\c*:)?(\i\c*|\*)))|\.)/)*((((child:)?((\i\c*:)?(\i\c*|\*)))|\.)|((attribute::|@)((\i\c*:)?(\i\c*|\*)))))*"/>
    </restriction>
  </simpleType>

```

6.2.2.3 XPath datatypes semantics

Semantics of the XPathType, XPathRefType, XPathSelectorType and XPathFieldType:

Name	Definition
xPathType	Basic datatype of XPath expressions as defined in (XPath). This datatype is based on the token type and serves as a base type for more restricted datatypes. This datatype is abstract and shall not be instantiated. Instead a derived datatype such as XPathRefType, XPathSelectorType or XPathFieldType must be instantiated.
xPathRefType	The datatype identifies one description element or attribute. It is based on XPathType and specifies in a pattern facet a subset (see Annex A) of XPath expressions as defined in (XPath). The regular expressions which are specified within the pattern facet are explained in detail in Annex F in XML Schema Part 2 (XML Schema).
xPathSelectorType	The datatype identifies a set of elements in the current description using an absolute XPath expression or relative to a context node. It is based on XPathType and specifies in a pattern facet a subset (see Annex A) of XPath expressions as defined in (XPath).
xPathAbsoluteSelectorType	The datatype identifies a set of elements in the current description using an absolute XPath expression. It is based on XPathSelectorType and specifies in a pattern facet a subset (see Annex A) of XPath expressions as defined in (XPath).
xPathFieldType	The datatype identifies a set of description elements and attributes in the current description relative to a context node. It is based on XPathType and specifies in a pattern facet a subset (see Annex A) of XPath expressions as defined in (XPath).

6.3 Unique Identifier

6.3.1 UniqueID datatype

6.3.1.1 Introduction

This datatype allows the identification of the multimedia or other media content under description. The identifier may be used to identify the content as a unique work being described (e.g., ISAN) or to identify its instances (e.g., SMPTE copy number).

6.3.1.2 UniqueID datatype syntax

```

<!-- ##### -->
<!-- Definition of UniqueID datatype (6.3.1) -->
<!-- ##### -->

<!-- Definition of UniqueID datatype -->
<complexType name="UniqueIDType">
  <simpleContent>
    <extension base="string">
      <attribute name="type" type="NMTOKEN" use="optional" default="URI"/>
      <attribute name="organization" type="NMTOKEN" use="optional"/>
      <attribute name="authority" type="NMTOKEN" use="optional"/>
      <attribute name="encoding" use="optional" default="text">
        <simpleType>
          <restriction base="string">
            <enumeration value="text"/>
            <enumeration value="base16"/>
            <enumeration value="base64"/>
          </restriction>
        </simpleType>
      </attribute>
    </extension>
  </simpleContent>
</complexType>

```

6.3.1.3 UniqueID datatype semantics

Semantics of the UniqueIDType:

Name	Definition
UniqueIDType	Describes the unique identification of a resource. An instance of this datatype contains a value (an identifier) that allows some resource to be identified. The identifying value can be either a textual or a binary value that is encoded in base16 or base64 format.
type	Describes the type of the identifier (e.g., URI, ISAN, ISWC, UMID, UPID). If no value is specified, the identifier is assumed to be a URI.
organization	Indicates the organization that defines the meaning of the identifier value (e.g., ISO, SMPTE) (optional).
authority	Indicates the authority that assigned this identifier value, for example a registration or maintenance authority (optional).
encoding	Indicates the encoding of the value (optional): • <i>text</i> – this identifier has a textual value. This is the default value. • <i>base16</i> – this identifier is encoded in a binary format. The binary value of the identifier is encoded into a textual representation by the base16 algorithm. • <i>base64</i> – this identifier is encoded in a binary format. The binary value of the identifier is encoded into a textual representation by the base64 algorithm.

6.3.1.4 UniqueID datatype examples (informative)

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 examples 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: 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.

6.4 Time description tools

6.4.1 Introduction

The time description is based on a restricted set of lexical expressions of the ISO 8601 standard. The lexical time expressions can represent the real world time (Time datatype) as well as the time as used in the AV data (mediaTime datatype). In both cases, time instances as well as time intervals can be described. The specification of fraction of seconds is not done following the ISO 8601 standard. Instead of using an arbitrary number of decimals (as in ISO 8601), fractions of seconds are specified by counting a predefined interval (of duration less than one second). This approach is widely used within AV media for instance in MPEG-4 video.

Furthermore, in view of the large amount of AV content that is temporally sampled, an efficient time representation specified as a number of sampling periods is also provided. The various entities used in the Time datatype (description of the real word time) are illustrated in Figure 4.

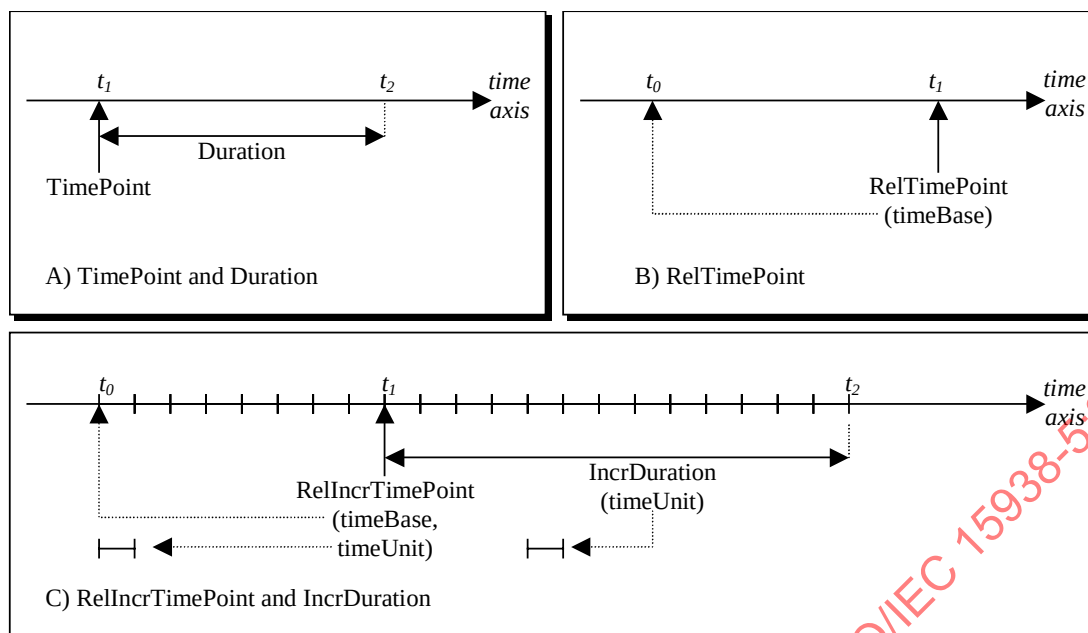


Figure 4 — Overview of Time Datatype entities

Figure 4(A) illustrates the simplest way to specify a temporal instant and a temporal interval. A time instant, t_1 , can be defined by a lexical representation using the TimePoint. An interval, $[t_1, t_2]$, can be defined by its starting point, t_1 , (using the TimePoint) and a duration, $t_2 - t_1$, also specified by a lexical representation. An alternative way to specify a time instant is shown in Figure 4(B). It relies on RelTimePoint. The instant, t_1 , is defined as a temporal offset with respect to a reference, t_0 , called timeBase.

Note: the goal of the RelTimePoint is to define a temporal instant, t_1 , and not an interval as the duration in Figure 4(A).

Finally, Figure 4(C) illustrates the specification of time using a predefined interval called timeUnit and counting the number of intervals. This specification is particularly efficient for periodic or sampled temporal signals. Since the strategy consists of counting timeUnits, the specification of a time instant shall be done relative to a timeBase (or temporal origin). In Figure 4(C), t_1 is defined with RelIncrTimePoint by counting 8 timeUnits (starting from t_0). An interval $[t_1, t_2]$, can also be defined by counting timeUnits. In Figure 4(C), IncrDuration is used to count 13 timeUnits to define the interval $[t_1, t_2]$.

As shown in Figure 4, the specification of the real world time can use five entities: timePoint, duration, RelTimePoint, RelIncrTimePoint and IncrDuration. The MediaTime involves the same notions: mediaTimePoint, mediaDuration, MediaRelTimepoint, MediaRelIncrTimePoint and MediaIncrDuration. Contrary to the Time entities the MediaTime entities do not specify time zone information.

6.4.2 Time datatype

6.4.2.1 Introduction

For the specification of time intervals the Time datatype is composed of two elements, the (start) time point and the duration. If only a time point is specified, the duration can be omitted.

6.4.2.2 Time datatype syntax

```
<!-- ##### -->
<!-- Definition of Time datatype (6.4.2) -->
<!-- ##### -->
```

```

<!-- Definition of Time datatype -->
<complexType name="TimeType">
  <sequence>
    <choice>
      <element name="TimePoint" type="mpeg7:timePointType" />
      <element name="RelTimePoint" type="mpeg7:RelTimePointType" />
      <element name="RelIncrTimePoint" type="mpeg7:RelIncrTimePointType" />
    </choice>
    <choice minOccurs="0" >
      <element name="Duration" type="mpeg7:durationType" />
      <element name="IncrDuration" type="mpeg7:IncrDurationType" />
    </choice>
  </sequence>
</complexType>

```

6.4.2.3 Time datatype semantics

Semantics of the TimeType:

Name	Definition
TimeType	Describes a time point or a time interval using one of the following descriptors to specify a time point and additionally in the case of a time interval one descriptor specifying the duration.
TimePoint	Describes a time point using Gregorian date and day time.
RelTimePoint	Describes a time point relative to a time base using a number of days and day time.
RelIncrTimePoint	Describes a time point relative to a time base counting time units.
Duration	Describes the duration of a time period according to days and day time (optional).
IncrDuration	Describes the duration of a time period counting time units (optional).

The Time datatype uses two different ways to describe a time point or duration:

- Using Gregorian date and time (TimePoint, RelTimePoint, Duration).
- By counting time units specified by an attribute (RelIncrTimePoint, IncrDuration). In the case of a description of depicted events in video segments for instance, these time units may correspond to the frame rate.

This enables to describe at what time the sequence is taken using Gregorian date, time and also to specify the time according to samples in time such as video frames.

6.4.3 TimePoint datatype

6.4.3.1 Introduction

The timePoint datatype specifies a time point according to the Gregorian dates and day time and the Time Zone (TZ). The format is based on the ISO 8601 norm. To reduce conversion problems only a subset of the ISO 8601 formats is used.

6.4.3.2 TimePoint datatype syntax

```

<!-- ##### -->
<!-- Definition of timePoint datatype (6.4.3) -->
<!-- ##### -->

<!-- Definition of timePoint datatype -->
<simpleType name="timePointType">
  <restriction base="mpeg7:basicTimePointType">
    <pattern value="(\-?\d+(\-\d{2}(\-\d{2}))?)? (T\d{2}(:\d{2}(:\d{2}
      (:\d+)?))?)?(F\d+)?((\-|\+)+\d{2}:\d{2})?" />
  </restriction>
</simpleType>

```

6.4.3.3 TimePoint datatype semantics

Semantic of the timePoint datatype:

Name	Definition
timePointType	Describes a time point. The timePoint datatype is derived from the basicTimePoint as defined in ISO/IEC 15938-2 by restricting the fractions to integer numbers instead of decimals. YYYY-MM-DDThh:mm:ss:nnnFNNN±hh:mm The following lexicals are used for digits of the corresponding date/time elements: • Y: Year, can be a variable number of digits, • M: Month, • D:Day, • h: hour, • m: minute, • s: second, • n: number of fractions, nnn can be any number between 0 and NNN-1 (NNN and with it nnn can have an arbitrary number of digits). • N: number of fractions of one second which are counted by nnn. NNN can have a arbitrary number of digits and is not limited to three. • ±hh:mm: the time zone according to the UTC specified by a positive or negative (±) number of hours (h) with two digits and number of minutes (m) also with two digits. If no time zone is specified a time zone of 00:00 is assumed. Also delimiters for the time specification (T) and the number of fractions of one second (F) are used.

Beside the specification of Gregorian Date and day time, counting the number (nnn) of fractions (FNNN) of a second can allow a higher precision than one second. If counting fractions is used, the number of fractions making up one second (FNNN) shall be specified along with the counted number of fraction (nnn). Thus the FNNN defines

the value range of the counted number of fractions (nnn) value. So the value range of 'nnn' is limited from 0 to FNNN-1. If no time zone is specified a time zone of 00:00 is assumed.

To support the current practice in AV material, the time expressions specify a fraction of a second by counting predefined fractions instead of a decimal expression used within the date and time datatypes of DDL. XML Schema-based time specifications (XML Schema) can be represented as follows:

For instance a timeInstant of T13:20:01.235 would be expressed using the timePoint datatype by T13:20:01:235F1000. According to the number of used decimal digits the number of fractions of one second are 1000 as specified in the timePoint datatype.

6.4.4 Duration datatype

6.4.4.1 Introduction

The duration datatype specifies the duration of a time period according to days and day time. The format is based on the ISO 8601 norm. To reduce converting problems only a subset of the ISO 8601 formats is used. Fractions of a second are specified according to the timePoint datatype.

6.4.4.2 Duration datatype syntax

```
<!-- ##### -->
<!-- Definition of duration datatype (6.4.4) -->
<!-- ##### -->

<!-- Definition of duration datatype -->
<simpleType name="durationType">
  <restriction base="mpeg7:basicDurationType">
    <pattern value="\-?P(\d+D)?(T(\d+H)?(\d+M)?(\d+S)?(\d+N)?)((\d+F)?((\d+|+)\d{2}:\d{2}Z))?" />
  </restriction>
</simpleType>
```

6.4.4.3 Duration datatype semantics

Semantic of the duration datatype:

Name	Definition
durationType	Describes a duration of a time period as a half open time interval with the closed end being at the beginning. The durationType is derived from the basicDurationType as defined in ISO/IEC 15938-2 by restricting the fraction (f) to integer numbers instead of decimals.

The duration for instance of a time segment can be specified using a number of days and time. The specification of fractions of a second (nN) is the same as in the case of timePoint. The scope of duration datatype is limited to days to avoid ambiguities depending on the date of the start time point if for instances describing the duration of a segment with "one month."

Note: the duration datatype can specify a difference in the TZ specification that indicates the difference in the time zone from the starting point to the end point (for instance using a mobile camera). In this case the value of the duration does not differ whether this shift in the time zone is specified or not. The duration always describes the time duration of a time period. The difference in the TZ is only used to map the resulting end time point of the period to the local time.

6.4.5 IncrDuration datatype

6.4.5.1 Introduction

To enable a simplified and efficient description of segment duration using a periodical time specification (e.g. periodic samples along the timeline) the IncrDuration datatype specifies a duration of such a segment by counting time units. Such a time unit can for example be a time increment between successive frames with respect to the world time when the sequence was recorded. The duration is then specified by the number of these time units.

6.4.5.2 IncrDuration datatype syntax

```
<!-- ##### -->
<!-- Definition of IncrDuration datatype (6.4.5) -->
<!-- ##### -->

<!-- Definition of IncrDuration datatype -->
<complexType name="IncrDurationType">
  <simpleContent>
    <extension base="integer">
      <attribute name="timeUnit" type="mpeg7:durationType" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

6.4.5.3 IncrDuration datatype semantics

Semantics of the IncrDurationType:

Name	Definition
IncrDurationType	Describes the duration of a temporal interval by counting the number of timeUnits. The IncrDuration datatype is derived from the simple type integer by adding an attribute specifying the timeUnit (of datatype duration) counted by the integer value.
timeUnit	Describes the duration of a small time interval called a time unit (optional if a timeUnit attribute is instantiated in an ancestor). The time interval is defined as a half open time interval with the closed end being at the beginning. The IncrDuration specifies a duration of a time interval by a multiple of this time unit.

6.4.6 RelTimePoint datatype

6.4.6.1 Introduction

The RelTimePoint datatype specifies a time point relative to a time base using a number of days and time. The format is based on the ISO 8601 norm. The specification is similar to the one used for duration datatype.

6.4.6.2 RelTimePoint datatype syntax

```
<!-- ##### -->
<!-- Definition of RelTimePoint datatypes (6.4.6) -->
<!-- ##### -->

<!-- Definition of timeOffsetType datatype -->
```



```

<simpleType name="timeOffsetType">
  <restriction base="mpeg7:basicDurationType">
    <pattern value="\-?P(\d+D)?(T(\d+H)?(\d+M)?(\d+S)?(\d+N)?)(\d+F)?((\+|-)
      |\+)\d{2}:\d{2}Z)?"/>
  </restriction>
</simpleType>

<!-- Definition of RelTimePointType datatype -->
<complexType name="RelTimePointType">
  <simpleContent>
    <extension base="mpeg7:timeOffsetType">
      <attribute name="timeBase" type="mpeg7:xPathRefType" use="optional"/>
    </extension>
  </simpleContent>
</complexType>

```

6.4.6.3 RelTimePoint datatype semantics

Semantics of the RelTimePointType:

Name	Definition
RelTimePointType	Describes a time point relative to a temporal origin called time base. It uses a time offset to define the interval between the time base and the current time point. The RelTimePoint datatype is derived from the timeOffset datatype by adding the specification of the timeBase from which a time point is specified.
timeBase	Selects an element of Time or timePoint datatype in the current description as time base or temporal origin (optional if a timeBase attribute is instantiated in an ancestor). The context node of the XPath expression is the element which contains this attribute.

Semantics of the timeOffsetType:

timeOffsetType	Describes a time offset with respect to a time base as it is specified by RelTimePoint datatype. The timeOffset datatype is derived from the basicDuration datatype as defined in ISO/IEC 15938-2 by restricting the fraction (f) to integer numbers instead of decimals.

6.4.7 RelIncrTimePoint datatype

6.4.7.1 Introduction

The RelIncrTimePoint datatype specifies a time point relative to a time base counting time units as already specified for the IncrDuration datatype.

6.4.7.2 RelIncrTimePoint datatype syntax

```

<!-- ##### -->
<!-- Definition of RelIncrTimePoint datatype (6.4.7) -->
<!-- ##### -->

<!-- Definition of RelIncrTimePointType datatype -->

```

```

<complexType name="RelIncrTimePointType">
  <simpleContent>
    <extension base="integer">
      <attribute name="timeUnit" type="mpeg7:durationType" use="optional"/>
      <attribute name="timeBase" type="mpeg7:xPathRefType" use="optional"/>
    </extension>
  </simpleContent>
</complexType>

```

6.4.7.3 RelIncrTimePoint datatype semantics

Semantics of the RelIncrTimePointType:

Name	Definition
RelIncrTimePointType	Describes a time point relative to a time base counting time units. The datatype is derived from the simple type integer by adding an attribute specifying the timeUnit (of datatype duration) counted by the integer value and by adding the specification of the timeBase from which a time point is specified relatively by using an expression of datatype IncrDuration.
timeUnit	Describes the duration of a small time interval called a time unit (optional if a timeUnit attribute is instantiated in an ancestor). The RelIncrTimePoint specifies a point in time relatively to a time base by a multiple of this time unit.
timeBase	Selects an element of Time or timePoint datatype in the current description as time base or temporal origin (optional if a timeBase attribute is instantiated in an ancestor). The context node of the XPath expression is the element which contains this attribute.

6.4.8 TimeProperty attribute group

6.4.8.1 Introduction

The timeProperty attribute group is used to specify time base and time unit values of time lines associated with the content for a subtree of the description.

6.4.8.2 TimeProperty attribute group syntax

```

<!-- ##### -->
<!-- Definition of timeProperty attribute group (6.4.8) -->
<!-- ##### -->

<!-- Definition of timePropertyGrp attribute group -->
<attributeGroup name="timePropertyGrp">
  <attribute name="timeBase" type="mpeg7:xPathRefType"
    use="optional"/>
  <attribute name="timeUnit" type="mpeg7:durationType"
    use="optional"/>
</attributeGroup>

```

6.4.8.3 TimeProperty attribute group semantics

Semantics of the timePropertyGrp:

Name	Definition
timeProperty	Describes properties of a time line associated with AV content. The element that contains an instance of timeProperty attributes and all its sub-elements in a description define a sub-tree of the description where the property values are valid. To all instances of elements of type RelTimePoint, RelIncrTimePoint and IncrDuration the values of timeBase and timeUnit attributes which are instantiated in their nearest ancestor element in the description shall apply, except in the case that these elements themselves specify values for timeBase and timeUnit.
timeBase	Describes the time base of a relative time definition by referring to an element of Time or timePoint datatype in the current description, as time base or temporal origin. The context node of the XPath expression is the element which contains this attribute.
timeUnit	Describes the duration of a small time interval called a time unit, to be used in the incremental specifications of relative time points and duration.

6.4.9 Time datatype examples (informative)

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

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

The following example describes the time at which the 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 in 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 following clause on `MediaTime` datatype. 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.

6.4.10 MediaTime datatype

6.4.10.1 Introduction

The `MediaTime` datatype specifies a notion of time encoded within the described media. For the specification of time intervals according to the time specifications stored in the media the `MediaTime` datatype is composed of two elements, the (start) time point and the duration. If a time point only is specified, the duration can be omitted.

6.4.10.2 MediaTime datatype syntax

```
<!-- ##### -->
<!-- Definition of MediaTime datatype (6.4.10) -->
<!-- ##### -->

<!-- Definition of MediaTime datatype -->
<complexType name="MediaTimeType">
  <sequence>
    <choice>
      <element name="MediaTimePoint" type="mpeg7:mediaTimePointType"/>
      <element name="MediaRelTimePoint" type="mpeg7:MediaRelTimePointType"/>
      <element name="MediaRelIncrTimePoint"
                type="mpeg7:MediaRelIncrTimePointType" />
    </choice>
    <choice minOccurs="0">
      <element name="MediaDuration" type="mpeg7:mediaDurationType"/>
      <element name="MediaIncrDuration" type="mpeg7:MediaIncrDurationType"/>
    </choice>
  </sequence>
</complexType>
```

6.4.10.3 MediaTime datatype semantics

Semantics of the `mediaTimeType`:

Name	Definition
<code>MediaTimeType</code>	Describes a media time point or a media time interval using the following descriptors for time specification.
<code>MediaTimePoint</code>	Describes a media time point using Gregorian date and day time
<code>MediaRelTimePoint</code>	Describes a media time point relative to a time base using a number of days and day time
<code>MediaRelIncrTimePoint</code>	Describes a media time point relative to a time base counting time units
<code>MediaDuration</code>	Describes the duration of a media time period according to days and day time (optional)

Name	Definition
MediaIncrDuration	Describes the duration of a media time period counting time units (optional)

The MediaTime datatype uses two different ways to describe a time point or period:

- Using Gregorian date and time (MediaTimePoint, MediaRelTimePoint, MediaDuration).
- By counting time units specified by attributes (MediaRelIncrTimePoint, MediaIncrDuration). In the case of video segments for example, these time units may correspond to the frame rate or to a clock reference encoded within the media.

This enables to specify the media time according to the format used for the time specifications in the media.

6.4.11 MediaTimePoint datatype

6.4.11.1 Introduction

The mediaTimePoint datatype specifies a time point according to the Gregorian dates and day time of time information which is contained in media data. The format is based on the ISO 8601 norm. To reduce converting problems only a subset of the ISO 8601 formats is used. Fractions of a second are specified according to the timePoint datatype.

6.4.11.2 MediaTimePoint datatype syntax

```
<!-- ##### -->
<!-- Definition of mediaTimePoint datatype (6.4.11) -->
<!-- ##### -->

<!-- Definition of mediaTimePoint datatype -->
<simpleType name="mediaTimePointType">
  <restriction base="mpeg7:basicTimePointType">
    <pattern value="(\-?\d+(\-\d{2}(\-\d{2}))?)?
                  (T\d{2}(:\d{2}(:\d{2}(:\d+))?)?)?(F\d+)?"/>
  </restriction>
</simpleType>
```

6.4.11.3 MediaTimePoint datatype semantics

Semantic of the mediaTimePoint datatype:

Name	Definition
mediaTimePointType	Describes a time stamp of the media using Gregorian date and day time without specifying the TZD (see also TimePoint datatype). YYYY-MM-DDThh:mm:ss:nnnFNNN

The following lexicals are used for digits of the corresponding date/time elements:

- Y: Year, can be a variable number of digits,
- M: Month,

Name	Definition
	• D:Day, • h: hour, • m: minute, • s: second, • n: number of fractions, nnn can be any number between 0 and NNN-1 (NNN and with it nnn can have an arbitrary number of digits). • N: number of fractions of one second which are counted by nnn. NNN can have a arbitrary number of digits and is not limited to three. Also delimiters for the time specification (T) and the number of fractions of one second are used (F).

Beside the specification of Gregorian Date and day time, counting the number (nnn) of fractions (FNNN) of a second can allow a higher precision than one second. If counting fractions is used, the number of fractions making up one second (FNNN) shall be specified along with the counted number of fraction (nnn). Thus the FNNN defines the value range of the counted number of fractions (nnn) value. So the value range of 'nnn' is limited from 0 to FNNN-1.

To support the current practice in AV material, the time expressions specify a fraction of a second by counting predefined fractions instead of a decimal expression used within the date and time datatypes of XML Schema (XML Schema). XML Schema-based time specifications can be represented as follows:

For instance a timeInstant of T13:20:01.235 would be expressed using the timePoint datatype by T13:20:01:235F1000. According to the number of used decimal digits the number of fractions of one second are 1000 as specified in the timePoint datatype.

6.4.12 MediaDuration datatype

6.4.12.1 Introduction

The mediaDuration datatype specifies the duration of a time period according to days and day time of a notion of time encoded in the media. The format is based on the ISO 8601 norm. To reduce converting problems only a subset of the ISO 8601 formats is used. Fractions of a second are specified according to the timePoint datatype.

6.4.12.2 MediaDuration datatype syntax

```
<!-- ##### -->
<!-- Definition of mediaDuration datatype (6.4.12) -->
<!-- ##### -->

<!-- Definition of mediaDuration datatype -->
<simpleType name="mediaDurationType">
  <restriction base="mpeg7:basicDurationType">
    <pattern value="\-?P(\d+D)?(T(\d+H)?(\d+M)?(\d+S)?(\d+N)?)(\d+F)?" />
  </restriction>
</simpleType>
```

6.4.12.3 MediaDuration datatype semantics

Semantics of the mediaDurationType:

Name	Definition
mediaDurationType	Describes the duration of a time interval according to days and day time of a notion of time encoded in the media without specifying a difference in the TZD (see also duration datatype). The time interval is defined as a half open time interval with the closed end being at the beginning. A simpleType representing a duration in time using a lexical representation of days (nD), time duration and a fraction specification (TnHnMnSnN) including the specification of the number of fractions of one second (nF): (-)PnDTnHnMnSnNnF The mediaDuration datatype is derived from basicDuration datatype by restriction. The regular expressions which are specified within the Pattern facet are explained in detail in Annex E of XML Schema Part 2 (XML Schema).

6.4.13 MediaIncrDuration datatype

6.4.13.1 Introduction

To enable a simplified and efficient description of a media segment duration using a periodical time specification (e.g. periodic samples along the timeline) the MediaIncrDuration datatype specifies a duration of such a segment by counting time units. Such a time unit can for example be the time increment of the timestamps of successive frames in a video stream. The duration is then specified by the number of these time units.

6.4.13.2 MediaIncrDuration datatype syntax

```

<!-- ##### -->
<!-- Definition of MediaIncrDuration datatype (6.4.13) -->
<!-- ##### -->

<!-- Definition of MediaIncrDuration datatype -->
<complexType name="MediaIncrDurationType">
  <simpleContent>
    <extension base="integer">
      <attribute name="mediaTimeUnit" type="mpeg7:mediaDurationType"
        use="optional"/>
    </extension>
  </simpleContent>
</complexType>

```

6.4.13.3 MediaIncrDuration datatype semantics

Semantics of the MediaIncrDurationType:

Name	Definition
MediaIncrDurationType	Describes the duration of a time period with respect to the media time counting time units (see also the IncrDuration datatype).

Name	Definition
mediaTimeUnit	Describes the duration of a small time interval. The time interval is defined as a half open time interval with the closed end being at the beginning. The MediaIncrDuration specifies a duration of a time interval by a multiple of this time unit (optional if a mediaTimeUnit attribute is instantiated in an ancestor).

6.4.14 MediaRelTimePoint datatype

6.4.14.1 Introduction

The MediaRelTimePoint datatype specifies a time point relating to a time base using a number of days and time according to media time stamps. The format is based on the ISO 8601 norm. The specification is similar to the one used for mediaDuration datatype.

6.4.14.2 MediaRelTimePoint datatype syntax

```
<!-- ##### -->
<!-- Definition of MediaRelTimePoint datatype (6.4.14) -->
<!-- ##### -->

<!-- Definition of mediaTimeOffset datatype -->
<simpleType name="mediaTimeOffsetType">
  <restriction base="mpeg7:basicDurationType">
    <pattern value="\-?P(\d+D)?(T(\d+H)?(\d+M)?(\d+S)?(\d+N)?)(\d+F)?" />
  </restriction>
</simpleType>

<!-- Definition of MediaRelTimePoint datatype -->
<complexType name="MediaRelTimePointType">
  <simpleContent>
    <extension base="mpeg7:mediaTimeOffsetType">
      <attribute name="mediaTimeBase" type="mpeg7:xPathRefType" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

6.4.14.3 MediaRelTimePoint datatype semantics

Semantics of the MediaRelTimePointType:

Name	Definition
MediaRelTimePointType	Describes a media time point relative to a time base without specifying a difference of the TZD (see also the RelTime datatype)
mediaTimeBase	Selects the time base of the relative time specification by referring to an element of MediaTime datatype or MediaLocator in the current description (optional if a mediaTimeBase is instantiated in an ancestor). The context node of the XPath expression is the element which contains this attribute. In the MediaLocator case, the origin of the timeline in the mediafile is used as time base if no MediaTime is specified in the MediaLocator. If a MediaTime is present in the instantiation of the MediaLocator as it is the case in general the time is measured relative to the media time point respectively media start time point.

Semantics of the MediaTimeOffsetType:

MediaTimeOffsetType	Describes a time offset with respect to a time base as it is specified by MediaRelTimePoint datatype. The MediaTimeOffset datatype is derived from the basicDuration datatype as defined in ISO/IEC 15938-2 without specifying a difference of the TZD and by restricting the fraction (f) to integer numbers instead of decimals.

6.4.15 MediaRelIncrTimePoint datatype

6.4.15.1 Introduction

The MediaRelIncrTimePoint datatype specifies a time point relative to a time base counting time units as already specified for the MediaIncrDuration datatype. If for instance an addressing of a frame by counting frames is needed MediaRelIncrTimePoint datatype can be used referring to the starting time of the shot or the whole video encoded within the media as a time base.

6.4.15.2 MediaRelIncrTimePoint datatype syntax

```
<!-- ##### -->
<!-- Definition of MediaRelIncrTimePoint datatype (6.4.15) -->
<!-- ##### -->

<!-- Definition of MediaRelIncrTimePoint datatype -->
<complexType name="MediaRelIncrTimePointType">
  <simpleContent>
    <extension base="integer">
      <attribute name="mediaTimeUnit" type="mpeg7:mediaDurationType"
        use="optional"/>
      <attribute name="mediaTimeBase" type="mpeg7:xPathRefType" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

6.4.15.3 MediaRelIncrTimePoint datatype semantics

Semantics of the MediaRelIncrTimePointType:

Name	Definition
MediaRelIncrTimePointType	Describes a media time point relative to a time base counting time units
mediaTimeUnit	Describes the duration of a small time interval called a time unit (optional if a mediaTimeUnit attribute is instantiated in an ancestor). The MediaRelIncrTimePoint specifies a point in time relatively to a time base by a multiple of this time unit.
mediaTimeBase	Selects the time base of the relative time specification by referring to an element of MediaTime datatype or MediaLocator datatype in the current description (optional if a mediaTimeBase is instantiated in an ancestor). In the second case the position of the MediaLocator with respect to the timeline of the media is used as time base. This is the beginning of the timeline if no mediaTime is specified within the locator. Otherwise as in the

Name	Definition
	first case the media time point respectively the media start time point is the time base. The context node of the XPath expression in this attribute is the element which contains this attribute.

6.4.16 MediaTimeProperty attribute group

6.4.16.1 Introduction

The MediaTimeProperty Attributes are used to specify time base and time unit values of time lines associated with the media for a subtree of the description.

6.4.16.2 MediaTimeProperty attribute group syntax

```

<!-- ##### -->
<!-- Definition of mediaTimeProperty attribute group (6.4.16) -->
<!-- ##### -->

<!-- Definition of mediaTimePropertyGrp attribute group -->
<attributeGroup name="mediaTimePropertyGrp">
  <attribute name="mediaTimeBase" type="mpeg7:xPathRefType"
    use="optional"/>
  <attribute name="mediaTimeUnit" type="mpeg7:mediaDurationType"
    use="optional"/>
</attributeGroup>

```

6.4.16.3 MediaTimeProperty attribute group semantics

Semantics of the MediaTimePropertyGrp:

Name	Definition
MediaTimePropertyGrp	Describes properties of a time line associated with AV media. The element that contains an instance of MediaTimeProperty attributes and all its sub-elements in a description define a subtree of the description where the property values are valid. To all instances of elements of type MediaRelTimePoint, MediaRelIncrTimePoint and MediaIncrDuration the values of mediaTimeBase and mediaTimeUnit attributes which are instantiated in their nearest ancestor element in the description shall apply, except in the case that these elements themselves specify values for mediaTimeBase and mediaTimeUnit.
mediaTimeBase	Describes the time base of a relative time definition by referring to an element of MediaTime datatype, MediaLocator datatype or mediaUri datatype in the current description, as time base or temporal origin. In the second case the position of the MediaLocator with respect to the time line of the media is used as time base. In the last case the origin of the time line in the mediafile is used as time base. The context node of the XPath expression in this attribute is the element which contains this attribute.
mediaTimeUnit	Describes the duration of a small time interval called a time unit, to be used in the incremental specifications of relative time points and duration.

6.4.17 MediaTime datatype examples (informative)

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 is 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 as follows:

```

<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 describe a single frame in a video sequence, the frame numbers or the timestamps can be used as follows:

```

<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]">PT18S21N</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>
```

The frame can be address 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 mediaTimeUnit and mediaTimeBase, the mediaTimeUnit and mediaTimeBase attributes can be used in VideoSegment as follows:

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

If the 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>
```

6.5 Media Locators

6.5.1 Introduction

To relate descriptions to content in general, a referencing mechanism called the MediaLocator is specified. The MediaLocator allows a referencing from the description to the media by two mechanisms: the MediaUri and the InlineMedia. The MediaUri is used to locate external media content whereas the InlineMedia includes the media content itself. The StreamID may be used to identify a stream within the located multimedia content.

For temporal media the TemporalSegmentLocator datatype can additionally instantiate the MediaTime or BytePosition to locate media chunks within this media content addressed by the basic MediaLocator datatype. For the location of images and frames a media time point or a BytePosition can be instantiated in addition to the mechanism supported in the basic MediaLocator datatype.

6.5.2 MediaLocator datatype

6.5.2.1 Introduction

The MediaLocator datatype serves as a basic datatype for locating described media in general. Based on the MediaLocator datatype more specific locators such as the ImageLocator datatype and the TemporalSegmentLocator datatype are derived.

The MediaLocator datatype can be composed of a MediaUri for locating a media element. The InlineMedia datatype enables an alternative method for locating media data, by embedding the media data into the description. In both cases a StreamID can identify a particular stream within the media consisting of multiple streams.

6.5.2.2 MediaLocator datatype syntax

```
<!-- ##### -->
<!-- Definition of MediaLocator datatype (6.5.2) -->
<!-- ##### -->

<!-- Definition of MediaLocator datatype -->
<complexType name="MediaLocatorType">
  <sequence>
    <choice minOccurs="0">
      <element name="MediaUri" type="anyURI"/>
      <element name="InlineMedia" type="mpeg7:InlineMediaType"/>
    </choice>
    <element name="StreamID" type="nonNegativeInteger" minOccurs="0"/>
  </sequence>
</complexType>
```

6.5.2.3 MediaLocator datatype semantic

Semantics of the MediaLocatorType:

Name	Definition
MediaLocatorType	Describes the location of media data in general. The location of media data can be specified either by reference to the external media data using the MediaUri element or by embedding the media data within the description as an InlineMedia instance. If neither the MediaUri nor the InlineMedia is instantiated, the reference to the external media data is assumed to be specified elsewhere in the description. Furthermore, a particular stream within the media data consisting of multiple streams can be specified using the StreamID element. If no StreamID is present, it is assumed either that the media data contains only a single stream, or that all streams in the media data are located simultaneously.
MediaUri	Describes the location of external media data (optional).
InlineMedia	Describes inline media data embedded in the description (optional).
StreamID	Identifies a particular stream contained in the media data consisting of multiple streams (optional).

6.5.3 InlineMedia datatype

6.5.3.1 Introduction

The InlineMedia datatype is used to embed binary-encoded multimedia data in a textual description. The multimedia data contained in an InlineMedia description may be either essence data or multimedia data representing other essence data - this depends on its context. The InlineMedia datatype enables the description of multimedia data located within the description itself, without having to refer to a location external to the description.

6.5.3.2 InlineMedia datatype syntax

```
<!-- ##### -->
<!-- Definition of InlineMedia datatype (6.5.3) -->
<!-- ##### -->

<!-- Definition of InlineMedia datatype -->
<complexType name="InlineMediaType">
  <choice>
    <element name="MediaData16" type="hexBinary"/>
    <element name="MediaData64" type="base64Binary"/>
  </choice>
  <attribute name="type" type="mpeg7:mimeType" use="required"/>
</complexType>
```

6.5.3.3 InlineMedia datatype semantics

Semantics of the InlineMediaType:

Name	Definition
InlineMediaType	Describes media data embedded in the description.
MediaData16	Describes binary media data encoded as a textual string in base-16 format.
MediaData64	Describes binary media data encoded as a textual string in base-64 format.
type	Describes the MIME type of media data.

6.5.3.4 InlineMedia datatype examples (informative)

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>
```

6.5.4 TemporalSegmentLocator datatype

6.5.4.1 Introduction

The TemporalSegmentLocator describes the location of temporal media data such as video and audio. The localization within a target temporal media data can be described using either the media time information or the byte location information.

6.5.4.2 TemporalSegmentLocator datatype syntax

```
<!-- ##### -->
<!-- Definition of TemporalSegmentLocator datatype (6.5.4) -->
<!-- ##### -->

<!-- Definition of TemporalSegmentLocator datatype -->
<complexType name="TemporalSegmentLocatorType">
  <complexContent>
    <extension base="mpeg7:MediaLocatorType">
      <choice minOccurs="0">
        <element name="MediaTime" type="mpeg7:MediaTimeType"/>
        <element name="BytePosition">
          <complexType>
            <attribute name="offset" type="nonNegativeInteger"
              use="required"/>
            <attribute name="length" type="positiveInteger"
              use="optional"/>
          </complexType>
        </element>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

6.5.4.3 TemporalSegmentLocator datatype semantics

Semantics of the TemporalSegmentLocatorType:

Name	Definition
TemporalSegmentLocatorType	Describes the location of temporal media data such as video and audio. A localization within a target temporal media data can be specified using either the MediaTime or the BytePosition element. If neither the MediaTime nor the BytePosition element is instantiated, an entire temporal media data is localized.
MediaTime	Describes the localization of a partial temporal segment within a given temporal media data using a time point or a time point and duration (optional). If the MediaUri or the InlineMedia is instantiated the time base of a relative time specification is set to the time line of the media by default and shall not be instantiated. If only the time point is specified, the segment is "open-ended," i.e., the end time of the segment corresponds to the end time of the located media data.
BytePosition	Describes the localization of a partial temporal segment within a given temporal media data using the numbers of byte offset and byte length (optional). If only the byte offset is specified, the segment is "open-ended," i.e., the end of the segment corresponds to the end of the located media data.

Name	Definition
offset	Describes the byte offset, i.e., the number of bytes to be skipped from the beginning of the media data specified by the MediaURI or InlineMedia element.
length	Describes the length of the segment data in number of bytes (optional).

6.5.5 ImageLocator datatype

6.5.5.1 Introduction

The ImageLocator describes the location of media data such as an image or a video frame. The localization within a target temporal media data can be described using either the media time information or the byte location information.

6.5.5.2 ImageLocator datatype syntax

```

<!-- ##### -->
<!-- Definition of ImageLocator datatype (6.5.5) -->
<!-- ##### -->

<!-- Definition of ImageLocator datatype -->
<complexType name="ImageLocatorType">
  <complexContent>
    <extension base="mpeg7:MediaLocatorType">
      <choice minOccurs="0">
        <element name="MediaTimePoint" type="mpeg7:mediaTimePointType"/>
        <element name="MediaRelTimePoint"
          type="mpeg7:MediaRelTimePointType"/>
        <element name="MediaRelIncrTimePoint"
          type="mpeg7:MediaRelIncrTimePointType"/>
        <element name="BytePosition">
          <complexType>
            <attribute name="offset" type="nonNegativeInteger"
              use="required"/>
            <attribute name="length" type="positiveInteger"
              use="optional"/>
          </complexType>
        </element>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

6.5.5.3 ImageLocator datatype semantics

Semantics of the ImageLocatorType:

Name	Definition
ImageLocatorType	Describes the location of media data such as an image or a video frame. The temporal position of the target media data within the temporal media data can be specified using one of the media time points (MediaTimePoint, the MediaRelTimePoint and the MediaRelIncrTimePoint) or the BytePosition.

Name	Definition
MediaTimePoint	Describes the temporal position of the spatial media data within the temporal media data, which is represented based on the MediaTimePoint datatype (optional).
MediaRelTimePoint	Describes the temporal position of the spatial media data within the temporal media data, which is represented based on the MediaRelTimePoint datatype (optional). If the MediaUri or the InlineMedia is instantiated the time base of the relative time specification is set to the time line of the media by default and shall not be instantiated.
MediaRelIncrTimePoint	Describes the temporal position of the spatial media data within the temporal media data, which is represented based on the MediaRelIncrTimePoint datatype (optional). If the MediaUri or the InlineMedia is instantiated the time base of the relative time specification is set to the time line of the media by default and shall not be instantiated.
BytePosition	Describes the position of the media data using the number of byte offset (optional).
offset	Describes the byte offset, i.e., the number of bytes to be skipped from the beginning of the media data specified by the MediaURI or InlineMedia element.
length	Describes the length of the image data in number of bytes (optional).

6.5.6 MediaLocator datatype examples (informative)

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 following 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 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>
```

7 Basic description tools

7.1 Introduction

This clause defines the basic tools, which are used as components for building other description tools. The basic tools are used in this part of the standard (ISO/IEC 15039-5) and also in the audio (see ISO/IEC 15938-4) and visual (see ISO/IEC 15938-3) parts of this standard. The tools defined in this clause are as follows.

Table 9 — Overview of Basic Description Tools

<i>Tool</i>	<i>Functionality</i>
Language Identification	Describes the language of a textual description or of the multimedia content. This standard uses the XML defined <code>xml:lang</code> attribute to identify the language used to write a textual description.
Text Annotation	Describes 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 represented 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	Describes language-independent terms for use in multimedia descriptions and schemes for classifying a domain using a set of such terms. 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	Describes 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	Describes geographical locations. The <code>Place</code> DS is used to describe real and fictional places.
Graphs and Relations	Describes 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	Describes hints for ordering descriptions
Affective Description	Describes an audience's affective response to multimedia content.
Phonetic Description	Describes the pronunciations of a set of words.

7.2 Language identification

7.2.1 Introduction

Two situations require language identification in this standard:

- **Description Language:** indicates the language in which a textual description is written.
- **Content Language:** indicates the language properties of the multimedia content being described.

The `xml:lang` attribute shall be used to indicate the language of a description, for example, to specify the language in which a textual annotation is written. When the language of the multimedia content is being indicated the DDL language datatype shall be used, for example, to specify the language of an audio track in a video.

7.2.2 `xml:lang` attribute

7.2.2.1 Introduction

The `xml:lang` attribute, which is defined in the XML 1.0 specification, specifies the language of a textual description.

Note: All description tools that need to indicate the language of their description shall declare this attribute explicitly and declare the attribute by referencing the definition for the `xml:lang` attribute.

7.2.2.2 `xml:lang` attribute syntax

The `xml:lang` attribute is defined in the XML 1.0 specification. While a definition of this attribute does not actually exist in DDL, the following definition shall be assumed to exist:

```
<attribute name="xml:lang" type="language"/>
```

7.2.2.3 `xml:lang` attribute semantics

Semantics of the `xml:lang` attribute:

Name	Definition
<code>xml:lang</code>	Describes the language of a textual description using the XML language attribute.

The possible values of the language code are as defined in the DDL language datatype (see ISO/IEC 15938-2). When the `xml:lang` attribute is specified for an XML element and that XML element contains other XML elements, the behavior is as defined in the XML 1.0 specification: "the intent declared with `xml:lang` is considered to apply to all attributes and content of the element where it is specified, unless overridden with an instance of `xml:lang` on another element within that content."

To use this attribute within a description tool, the following form shall be used:

```
<attribute ref="xml:lang" use="optional"/>
```

7.2.2.4 xml:lang attribute examples (informative)

The following example shows the use of the `xml:lang` attribute for describing the language of text in a text annotation. In this example, the `TextAnnotation` describes the language of the description as English ("en"). This applies to the text of the elements contained within the `TextAnnotation`. For example, the language of the keyword "dog" is English. However, since `xml:lang` gives a value of "fr" for keyword "chien," the language for this keyword is French.

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

7.2.3 language datatype

7.2.3.1 Introduction

The `xsd:language` datatype, which is a built-in simple datatype in DDL (see ISO/IEC 15938-2), represents natural language identifiers.

7.2.3.2 language datatype syntax

The syntax of this datatype is defined in the DDL specification.

7.2.3.3 language datatype semantics

The `xsd:language` identifier contained within the language code shall be the primary method of designating the language of the target multimedia content. Appending a sub-code to indicate a country is an optional, secondary method for further restricting the language.

7.2.3.4 language datatype examples (informative)

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 following example describes a preferred language of English.

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

7.3 Textual annotation

7.3.1 Introduction

This subclause specifies tools for representing textual description and textual annotations.

7.3.2 Textual datatypes

7.3.2.1 Introduction

The `TextualBase` datatype provides a base from which other kinds of textual description tools are derived. It can also be used in the definitions of tools that contain textual descriptions. An optional attribute, `xml:lang`, is used to specify the language of the contained text.

The Textual datatype, derived from the TextualBase datatype, is a non-abstract type that is used for simple textual descriptions.

7.3.2.2 Textual datatypes syntax

```
<!-- ##### -->
<!-- Definition of Textual datatypes (7.3.2) -->
<!-- ##### -->

<!-- Definition of TextualBase datatype -->
<complexType name="TextualBaseType" abstract="true">
  <simpleContent>
    <extension base="string">
      <attribute ref="xml:lang" use="optional"/>
      <attribute name="phoneticTranscription" use="optional">
        <simpleType>
          <list itemType="mpeg7:PhoneType"/>
        </simpleType>
      </attribute>
      <attribute name="phoneticAlphabet" type="mpeg7:phoneticAlphabetType"
        use="optional" default="sampa"/>
    </extension>
  </simpleContent>
</complexType>

<!-- Definition of Textual datatype -->
<complexType name="TextualType">
  <simpleContent>
    <extension base="mpeg7:TextualBaseType"/>
  </simpleContent>
</complexType>
```

7.3.2.3 Textual datatypes semantics

Semantics of the TextualBaseType:

Name	Definition
TextualBaseType	Abstract base type for textual descriptions. This type includes the xml:lang attribute for identifying the language in which the description is written and can contain a phonetic transcription of the text
xml:lang	Indicates the language in which the textual description is written (optional). If no value is specified for this attribute, the value of the xml:lang attribute is inherited from the closest containing element that specifies a value for xml:lang.
phoneticTranscription	Describes a phonetic transcription of the text using a phonetic alphabet. The transcription is represented as a list of phones (i.e. PhoneType). The definition for PhoneType is given in ISO/IEC 15938-4.
phoneticAlphabet	Describes the phonetic alphabet that is used for the transcription in phoneticTranscription, e.g. IPA (International Phonetic Association alphabet). The specification of the phoneticAlphabetType is given in ISO/IEC 15938-4.

Semantics of the TextualType:

Name	Definition
TextualType	Describes a textual description. TextualType is a concrete type derived from the abstract TextualBaseType. This type includes the xml:lang attribute for identifying the language in which the description is written and can contain a phonetic transcription of the text

7.3.2.4 Textual datatypes example (informative)

The following example description excerpt shows the use of TextualType for describing the free text in English: "The Giants win the pennant."

```
<Textual xml:lang="en">
  The Giants win the pennant.
</Textual>
```

7.3.3 TextAnnotation datatype

7.3.3.1 Introduction

The TextAnnotation datatype represents a textual annotation using several different forms: free text, structured annotation, keyword annotation, or dependency structure. It can contain multiple forms of an annotation including translations in multiple languages, or a combination of both structured and unstructured descriptions of the same annotation.

7.3.3.2 TextAnnotation datatype syntax

```
<!-- ##### -->
<!-- Definition of TextAnnotation datatype (7.3.3) -->
<!-- ##### -->

<!-- Definition of TextAnnotation datatype -->
<complexType name="TextAnnotationType">
  <choice minOccurs="1" maxOccurs="unbounded">
    <element name="FreeTextAnnotation" type="mpeg7:TextualType"/>
    <element name="StructuredAnnotation" type="mpeg7:StructuredAnnotationType"/>
    <element name="DependencyStructure" type="mpeg7:DependencyStructureType"/>
    <element name="KeywordAnnotation" type="mpeg7:KeywordAnnotationType"/>
  </choice>
  <attribute name="relevance" type="mpeg7:zeroToOneType" use="optional"/>
  <attribute name="confidence" type="mpeg7:zeroToOneType" use="optional"/>
  <attribute ref="xml:lang"/>
</complexType>
```

7.3.3.3 TextAnnotation datatype semantics

Semantics of the TextAnnotationType:

Name	Definition
TextAnnotationType	Describes a textual annotation using different possible formats: free text, dependency-structured text, structured text annotation, or keywords.

Name	Definition
	An arbitrary number of textual descriptions can be grouped together. The grouping implies that the annotations are variations – in different formats or languages – of each other.
FreeTextAnnotation	Describes a free text annotation.
StructuredAnnotation	Describes a thematically structured annotation.
DependencyStructure	Describes a textual annotation with a syntactic parse based on dependency structures.
KeywordAnnotation	Describes a keyword annotation.
relevance	Identifies the relevance of the text annotation to item being described (optional). Relevance is specified on a zero-to-one scale, where zero indicates no relevance and one indicates highest relevance.
confidence	Indicates the confidence in the correctness of the text annotation (optional). Confidence is specified on a zero-to-one scale, where zero indicates no confidence and one indicates highest confidence.
xml:lang	Indicates the language of the textual annotation.

7.3.3.4 TextAnnotation datatype examples (informative)

The following example shows the use of the TextAnnotation datatype for annotating a video depicting the events "Tanaka throws a small ball to Yamada. Yamada catches the ball with his left hand." In this example, alternative expressions are given for describing the events as follows:

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

```
<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>
          </StructuredAnnotation>
        </TextAnnotation>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```



```

        </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>

```

7.3.4 StructuredAnnotation datatype

7.3.4.1 Introduction

The StructuredAnnotation datatype represents an annotation structured in terms of actions, animate object (people and animals), objects, action, places, time, purposes, and manner.

7.3.4.2 StructuredAnnotation datatype syntax

```

<!-- ##### -->
<!-- Definition of StructuredAnnotation datatype (7.3.4) -->
<!-- ##### -->

<!-- Definition of StructuredAnnotation datatype -->
<complexType name="StructuredAnnotationType">
    <sequence>
        <element name="Who" type="mpeg7:TermUseType"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="WhatObject" type="mpeg7:TermUseType"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="WhatAction" type="mpeg7:TermUseType"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="Where" type="mpeg7:TermUseType"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="When" type="mpeg7:TermUseType"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="Why" type="mpeg7:TermUseType"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="How" type="mpeg7:TermUseType"
            minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute ref="xml:lang" use="optional"/>
</complexType>

```

7.3.4.3 StructuredAnnotation datatype semantics

Semantics of the StructuredAnnotationType:

Name	Definition
StructuredAnnotationType	Describes a structured textual annotation in terms of who (people and animals), what object, what action, where (places), when (time), why (purpose), and how.
Who	Describes animate objects or beings (people and animals) or legal persons (organizations and person groups) using either free text or a term from a classification scheme.
WhatObject	Describes inanimate objects using either free text or a term from a classification scheme.
WhatAction	Describes actions using either free text or a term from a classification scheme.
Where	Describes a place using either free text or a term from a classification scheme.
When	Describes a time using either free text or a term from a classification scheme.
Why	Describes a purpose or reason using either free text or a term from a classification scheme.
How	Describes a manner using either free text or a term from a classification scheme.

7.3.4.4 StructuredAnnotation datatype examples (informative)

The following example description excerpt shows the use of the StructuredAnnotation datatype for describing a structured annotation for a video in which "Mr. Jones tips the canoe."

```
<StructuredAnnotation>
  <Who>
    <Name xml:lang="en">Mr. Jones</Name>
  </Who>
  <WhatObject>
    <Name xml:lang="en"> Canoe </Name>
  </WhatObject>
  <WhatAction>
    <Name xml:lang="en"> Tips </Name>
  </WhatAction>
</StructuredAnnotation>
```

7.3.5 KeywordAnnotation datatype

7.3.5.1 Introduction

The KeywordAnnotation datatype represents an annotation as a set of keywords.

7.3.5.2 KeywordAnnotation datatype syntax

```
<!-- ##### -->
<!-- Definition of KeywordAnnotation datatype (7.3.5) -->
<!-- ##### -->
```

```

<!-- Definition of KeywordAnnotation datatype -->
<complexType name="KeywordAnnotationType">
  <sequence>
    <element name="Keyword" minOccurs="1" maxOccurs="unbounded">
      <complexType>
        <simpleContent>
          <extension base="mpeg7:TextualType">
            <attribute name="type" use="optional" default="main">
              <simpleType>
                <restriction base="NMTOKEN">
                  <enumeration value="main"/>
                  <enumeration value="secondary"/>
                  <enumeration value="other"/>
                </restriction>
              </simpleType>
            </attribute>
          </extension>
        </simpleContent>
      </complexType>
    </element>
  </sequence>
  <attribute ref="xml:lang" use="optional"/>
</complexType>

```

7.3.5.3 KeywordAnnotation datatype semantics

Semantics of the KeywordAnnotationType:

Name	Definition
KeywordAnnotationType	Describes an annotation represented as a set of keyword phrases.
Keyword	Describes one keyword. A keyword can be a single word or an entire phrase made up of multiple words. For example, "President of the United States" can be treated as a keyword.
type	Indicates the type of keyword. The following values shall be used: <i>main</i> – The keyword is the main or primary keyword. This designates that this keyword is considered important or characteristic. <i>secondary</i> – The keyword is an alternative keyword. This designates that this keyword is considered less important or relevant than a "main" keyword. <i>other</i> – The keyword is an other keyword. This designates that this keyword is considered less important or relevant than either a "main" or a "secondary" keyword.
xml:lang	Identifies the language in which the keywords are written.

7.3.5.4 KeywordAnnotation datatype examples (informative)

The following example shows the use of the KeywordAnnotation datatype for 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>United States of America</Keyword>
          <Keyword>Presidential election</Keyword>
        </KeywordAnnotation>
      </TextAnnotation>
    </Video>
  </MultimediaContent>
</Description>
</Mpeg7>

```

7.3.6 DependencyStructure datatype

7.3.6.1 Introduction

The DependencyStructure datatype represents the structure of a text annotation based on the syntactic dependency structure of the grammatical elements making up a sentence. It is based on a linguistic theory called dependency grammar, which explains a sentence's grammatical structure in terms of the dependencies between its elements. A syntactic constituent, such as a sentence and a noun phrase, contains a governor and a number of dependents whose number and type is determined by properties of the governor. For example, in "Tom likes fruits.", the verb, *like*, is the governor, and its dependents are the nouns, *Tom* and *fruits*. The governor *like* requires both a subject *Tom* and an object *fruits*. The two dependents make different contributions to the meaning of the whole - the subject *Tom* contributes the 'like-er' and the object *fruits* contributes the 'like-ee'. What distinguishes a governor is that it allows its dependents to appear. For example, in *Mary bought red apples*, *red* is linked to the rest of the sentence via its governor *apples*. *Without apples, red could not appear in the sentence*.

Because dependency structure can be translated easily into a formal logical representation of the semantics of a sentence, it is used in some systems for understanding and translating languages. The DependencyStructure datatype supports text annotation based retrieval where inter-element relationships such as "A gives B to C" are specified, which cannot be expressed solely by Boolean operators (AND/OR) and keywords. The dependency structure of a sentence is represented using a dependency tree as shown in Figure 5. Nodes of the tree are labeled with all the structural primitives of the sentence, which usually correspond to words. Each edge represents a dependency between the connected primitives. The edges can be labeled with names of the dependency relations. The tree consists of subtrees; each of which contains a governor and a set of dependents. In the diagram, the dependents at the end of the arrows depend on the governor at the beginning of the arrow.

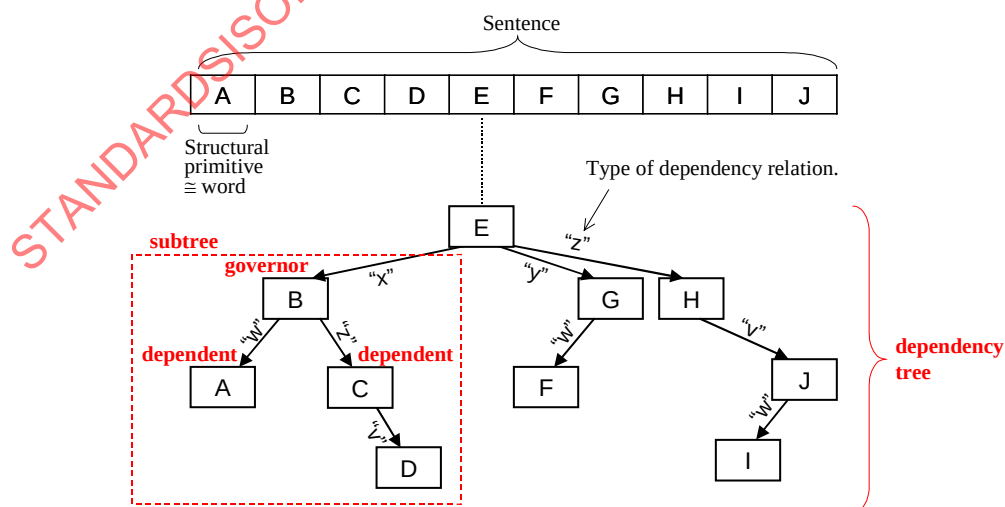


Figure 5 — Illustration of the dependency structure among the governors and dependents of a sentence

The syntactic structure representation used in the DependencyStructure datatype is based on the following:

- The structural primitive is called a word-form.
- Each word-form depends directly on only one other word-form.
- One and only one word-form is independent—i.e. has no governor – in a sentence. (Node labeled "E" in Figure 5 is independent.). All others depend directly on some word-form.
- A sentence is a complete textual unit, which is the largest unit of syntactic analysis.

The `DependencyStructure` datatype represents the dependency tree of a sentence as a nested set of XML elements. A `Phrase` element, whose content model is specified by the `DependencyStructurePhraseType`, represents one subtree in the dependency tree and contains a single `Head` (or `CompoundHead`) and zero or more child `Phrases`. A phrase is defined as a syntactic constituent consisting of several word-forms that can be structured based on the dependency relations and build up one subtree. A head is a word-form corresponding to the top most node in the subtree. Each child phrase occurring in the parent phrase represents a 'sub'-subtree and there are dependency relations among word-forms that are heads of parent and child phrases. For example, in phrase *the book on the desk in the room*, *in* is a head of the prepositional phrase *in the room*; this *in* is a dependent of *desk*, which is the head of a larger phrase *the desk in the room*; that *in* is the dependent of *on*, forming a larger prepositional phrase *on the desk in the room*, which is itself a dependent of *book*.

A dependency relation exists between a head word-form of a parent phrase, i.e., a governor, and a head word-form of its child phrase, i.e., a dependent. In the following subclauses, this relationship is expressed as "a [child] *phrase* depends on a *head* [of its parent phrase]."

In some cases, the structure of a phrase in a sentence cannot be represented by dependency. For example, in *Tom and Mary walk*, *Tom and Mary* forms a coordination structure in which word-forms are joined by means of a coordinating conjunction and none of them can be said to depend on any other. The `NonDependencyStructurePhraseType` is defined to represent a non-dependency structure of such a phrase. Being instantiated as a `CompoundHead` element, the type can encapsulate the non-dependency structure and make it a node of the dependency tree.

7.3.6.2 `DependencyStructure` datatype syntax

```
<!-- ##### -->
<!-- Definition of Dependency Structure datatypes (7.3.6) -->
<!-- ##### -->

<!-- Definition of DependencyStructure datatype -->
<complexType name="DependencyStructureType">
  <sequence>
    <element name="Sentence"
      type="mpeg7:DependencyStructurePhraseType"
      minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
  <attribute ref="xml:lang" use="optional"/>
</complexType>

<!-- Definition of DependencyStructurePhrase datatype -->
<complexType name="DependencyStructurePhraseType">
  <sequence>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="Quotation" type="mpeg7:DependencyStructurePhraseType"/>
      <element name="Phrase" type="mpeg7:DependencyStructurePhraseType"/>
    </choice>
    <choice>
      <element name="Head" type="mpeg7:WordFormType"/>
      <element name="CompoundHead" type="mpeg7:NonDependencyStructurePhraseType"/>
    </choice>
  </sequence>
  <attribute ref="xml:lang" use="optional"/>
</complexType>
```

```

        type="mpeg7:NonDependencyStructurePhraseType"/>
    </choice>
    <choice minOccurs="0" maxOccurs="unbounded">
        <element name="Quotation" type="mpeg7:DependencyStructurePhraseType"/>
        <element name="Phrase" type="mpeg7:DependencyStructurePhraseType"/>
    </choice>
</sequence>
<attribute name="id" type="ID" use="optional"/>
<attribute name="equal" type="IDREF" use="optional"/>
<attribute name="operator" use="optional">
    <simpleType>
        <union memberTypes="mpeg7:dependencyOperatorType
mpeg7:termReferenceType"/>
    </simpleType>
</attribute>
<attribute name="functionWord" type="string" use="optional"/>
<attribute name="synthesis" type="NMTOKEN" use="optional" fixed="dependency"/>
</complexType>

<!-- Definition of WordForm datatype -->
<complexType name="WordFormType">
    <simpleContent>
        <extension base="string">
            <attribute name="terms" use="optional">
                <simpleType>
                    <list itemType="mpeg7:termReferenceType"/>
                </simpleType>
            </attribute>
            <attribute name="id" type="ID" use="optional"/>
            <attribute name="equal" type="IDREF" use="optional"/>
            <attribute name="type" use="optional">
                <simpleType>
                    <list>
                        <simpleType>
                            <union>
                                <simpleType>
                                    <restriction base="NMTOKEN">
                                        <enumeration value="noun"/>
                                        <enumeration value="pronoun"/>
                                        <enumeration value="adjective"/>
                                        <enumeration value="verb"/>
                                        <enumeration value="adverb"/>
                                        <enumeration value="conjunction"/>
                                        <enumeration value="preposition"/>
                                        <enumeration value="postposition"/>
                                        <enumeration value="article"/>
                                        <enumeration value="interjection"/>
                                    </restriction>
                                </simpleType>
                                <simpleType>
                                    <restriction base="mpeg7:termReferenceType"/>
                                </simpleType>
                            </union>
                        </simpleType>
                    </list>
                </simpleType>
            </attribute>
            <attribute name="baseForm" type="string" use="optional"/>
        </extension>
    </simpleContent>

```

```

</complexType>

<!-- Definition of NonDependencyStructurePhrase datatype -->
<complexType name="NonDependencyStructurePhraseType">
  <choice minOccurs="2" maxOccurs="unbounded">
    <element name="Quotation" type="mpeg7:DependencyStructurePhraseType"/>
    <element name="Phrase" type="mpeg7:DependencyStructurePhraseType"/>
    <element name="Word" type="mpeg7:WordFormType"/>
    <element name="NonDependencyPhrase"
      type="mpeg7:NonDependencyStructurePhraseType"/>
  </choice>
  <attribute name="id" type="ID" use="optional"/>
  <attribute name="equal" type="IDREF" use="optional"/>
  <attribute name="synthesis" use="optional" default="unspecified">
    <simpleType>
      <restriction base="NMTOKEN">
        <enumeration value="unspecified"/>
        <enumeration value="coordination"/>
      </restriction>
    </simpleType>
  </attribute>
</complexType>

<!-- Definition of dependencyOperator datatype -->
<simpleType name="dependencyOperatorType">
  <restriction base="NMTOKEN">
    <enumeration value="subject"/>
    <enumeration value="object"/>
    <enumeration value="indirectObject"/>
    <enumeration value="predicate"/>
    <enumeration value="expletive"/>
    <enumeration value="nonRestrictive"/>
  </restriction>
</simpleType>

```

7.3.6.3 DependencyStructure datatype semantics

Semantics of the DependencyStructureType:

Name	Definition
DependencyStructureType	Describes a textual annotation with a syntactic parse tree consisting of dependency relations. A dependency relation holds between a governor (or head) and its dependents. The parse tree, called a dependency tree, is represented by including dependent elements within their governing element.
Sentence	Describes a phrase that is syntactically complete and does not depend on any other elements. This is the top-level phrase corresponding to the whole dependency tree.
xml:lang	Indicates the language used in this DependencyStructure.

Semantics of the DependencyStructurePhraseType:

Name	Definition
DependencyStructurePhraseType	A phrase, which represents one subtree in the dependency tree. A phrase must contain either a <i>head</i> , which is the representative word-form of the whole phrase, or a <i>compound head</i> , which is a head composed of multiple sub-elements (phrases or word-forms) without any dependency relations among them. A phrase may recursively contain child phrases, which depend on the head.
Quotation	Describes a directly quoted discourse (e.g. "He said 'I am hungry. Let's go to lunch.'" but not "He said that he was hungry."). This element can appear zero or more times in a phrase.
Phrase	Describes a child phrase that depends on the head of this phrase. This element can appear zero or more times in the phrase.
Head	Describes the head of the phrase. Either a Head or CompoundHead must be instantiated in each phrase.
CompoundHead	Describes the compound head of the phrase. Either a Head or CompoundHead must be instantiated in each phrase.
id	Identifies the phrase. It must be unique and is used as a key for linking coreferents of this phrase via the equal attribute.
equal	Describes a reference to a word-form or phrase that has the same referent with this phrase. The value is the id of the coreferent.
operator	Indicates the type of dependency relationship between this phrase and its governing head, which is the Head or CompoundHead element occurring in the parent phrase of this phrase. The type shall be selected from the enumerated list of the dependencyOperatorType or a relevant term from a classification scheme. Other values that are datatype-valid with respect to mpeg7:termReferenceType are reserved.
functionWord	Indicates a function word representing the surface syntactic relationship between this phrase and the governing head. This and the operator provide complementary information about the dependency relationship. For example, the sentence "Tom talks to Mary," in which the function word is "to," can be structured as; <pre> <Sentence> <Phrase operator="subject"> <Head>Tom</Head> </Phrase> <Head baseForm="talk">talks</Head> <Phrase functionWord="to"> <Head>to</Head> <Phrase> <Head>Mary</Head> </Phrase> </Phrase> </Sentence> </pre>
synthesis	Indicates the type of synthesis among the child entities of this phrase. The value is fixed to be <i>dependency</i>: <i>dependency</i> – The synthesis is dependency. Each child element, except for one, depends on the Head/CompoundHead.

Semantics of the WordFormType:

Name	Definition
WordFormType	Describes a structural primitive of a sentence, called a word-form.
terms	References relevant terms in a classification scheme. Relevant terms are those capturing the meaning of the word-form being described.
id	Identifies the word-form. It must be unique and is used as a key for linking coreferents of this word-form via the equal attribute.
equal	References another word-form or phrase which refers to the same thing as this word-form. The value is the id of the coreferent. A word-form or phrase that refers to the same thing as another word-form or phrase is called a coreferent; for example, a pronoun and its antecedent referent. For example, in the sentence "I smiled to myself," both the terms for "I" and "myself" are coreferents.
type	Indicates the type of the word-form; that is, the part of speech represented by the word-form. It should be selected from an enumerated list including <i>noun</i>, <i>pronoun</i>, <i>adjective</i>, <i>verb</i>, <i>adverb</i>, <i>conjunction</i>, <i>preposition</i>, <i>postposition</i>, <i>article</i> and <i>interjection</i>, or some relevant term in a classification scheme. When the word-form includes two or more morphemes, this attribute is allowed to have a list of the values. Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.
baseForm	Indicates the lexical base form or uninflected form of this word-form, i.e., the dictionary look-up form of the word. For example, the base form of past tense English verb "struck" is "strike." The base form can be used as a key for query matching or dictionary reference. It should be noted that what constitutes a base form may depend on the linguistic model used for a given language.

Semantics of the NonDependencyStructurePhraseType:

Name	Definition
NonDependencyStructurePhraseType	Describes a syntactic constituent that consists of several sub-elements with non-dependency relationship such as coordination. Each such element shall be represented as a separate element in the content type. Furthermore, the order of these elements reproduces the original word order.
Quotation	Describes directly quoted discourse. This element can appear zero or more times.
Phrase	Describes a sub-phrase that forms one dependency subtree. This element can appear zero or more times.
Word	Describes a word-form. This element can appear zero or more times.
NonDependencyPhrase	Describes a sub-phrase that is composed of sub-elements with non-dependency relations among them. This element can appear zero or

Name	Definition
	more times.
id	Identifies this phrase. It must be unique and is used as a key for linking coreferents of this phrase via the equal attribute.
equal	References another word-form or another phrase that has the same referent with this phrase. The value is the id of the coreferent.
synthesis	Indicates the type of synthesis among the child elements of the phrase. • <i>coordination</i> – The synthesis is coordination. For example, synthesis type of the underlined phrase in the following sentence: "Fishes <u>and</u> turtles in water," is coordination, i.e., "Fishes" and "turtle" has a coordinate relationship. There is no dependency structure in coordination synthesis. • <i>unspecified</i> – An undefined value allowing any kind of synthesis other than dependency or coordination, for instance, compound prepositions (ex., "in front of") and compound proper names. This is the default value for the synthesis attribute.

Semantics of the dependencyOperatorType:

Name	Definition
dependencyOperatorType	Indicates the type of relationship between a governor and a dependent. It can have one of the following values. • <i>subject</i> – The dependent is grammatical constituent about which something is predicated in the sentence. • <i>object</i> – The dependent is a grammatical constituent that is acted upon. • <i>indirectObject</i> – The dependent is a syntactic indirect object. For example, the underlined phrase in the following sentence is an indirect object: E.g. I gave <u>her</u> a bouquet. • <i>predicate</i> – The dependent is a predication of the subject or object of the governing verb. For example, each underlined phrase in the following sentences is a predicate: Tom is a <u>doctor</u>. Mary is <u>cute</u>. Bill made John a <u>lawyer</u>. Mary is considered <u>stupid</u>. Sue looks <u>confident</u>. • <i>expletive</i> – The dependent is a word or phrase conveying no independent meaning but added to fill out a sentence or metrical line. For example, each underlined phrase in the following sentences is an expletive: <u>It</u> is nice to meet you. <u>There</u> is a book on the table. • <i>nonrestrictive</i> – The dependent does not limit the reference of a governing word or phrase. For example, the underlined phrases in the following sentences are nonrestrictive phrases: Japan, <u>where I lived for decades</u>. The king, <u>whom I know very well</u>, is now quite ill. Other values that are datatype-valid with respect to mpeg7:termReferenceType are reserved.

7.3.6.4 DependencyStructure datatype examples (informative)

The following rules specify the dependency relations in English:

<i>Governor</i>	<i>Dependent</i>	<i>Operator/FunctionWord</i> (of DependencyStructure datatype)	<i>Example</i> (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)
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

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

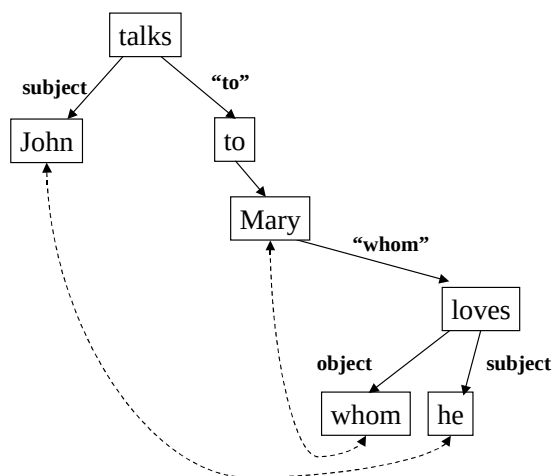


Figure 6 — Example of dependency structure of the sentence: "John talks to Mary whom he loves"

The solid arrows in Figure 6 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:

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

```
<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>
              <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>
              </Phrase>
            </Sentence>
          </DependencyStructure>
        </TextAnnotation>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```

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

```

7.4 Classification Schemes and Terms

7.4.1 Introduction

This subclause specifies tools for representing classification schemes (ClassificationScheme DS), defining terms inside a classification scheme (TermDefinition DS), and for using terms inside descriptions (TermUse and ControlledTermUse datatypes).

This standard defines classification schemes and terms as follows.

- **Classification Scheme:** A set of standard **terms** that describes some domain. A classification scheme may organize the terms it contains with a set of **term relations**.

Note: Depending on the field, a classification scheme may be called by different names. In information retrieval, it is often called a thesaurus or a controlled vocabulary, in the knowledge representation community it is called an ontology, and in some scientific fields it may be called a taxonomy. In the rest of this subclause, classification scheme is used to cover all of these senses; however, the use of classification schemes in this standard is closest to "thesaurus."

- **Term.** A term represents one well-defined concept in the domain covered by the classification scheme. A term has an **identifier** that uniquely identifies a term, a **name** that may be displayed to a user or used as a search term in a target database, and a **definition** that describes the meaning of the term.
- **Term Relation.** A relation between two terms in a classification scheme, such as whether one term includes the meaning of another term. In most cases, the relation between terms indicates whether one term is "narrower than" or "broader than" another term in meaning. When terms are organized this way, they form a classification hierarchy. The following relationships between two terms shall be used:

Table 10 — Overview of Relations between Terms

<i>Term Relation</i>	<i>Definition</i>
Use	The use of the related term is (nearly) synonymous, but the use of the related term is preferred to the current term.
Used for	The use of the current term is preferred to the use of the (nearly) synonymous related term.
Broader Term	The meaning of the related term is more general than the current term.
Narrower Term	The meaning of the related term is more specific than the current term.
Related Term	The related term and current term are associated in some way other than those covered by one of the four relations above.

- **Term Use.** A term is **used** or **referenced** inside a description as a label.
- **Controlled Term Use.** A term used in a description is **controlled** by requiring the term being employed to be uniquely identified. This is accomplished by referencing a term in a classification scheme.
- **Term Reference.** A reference to a term in a classification scheme.

7.4.2 ClassificationScheme DS

7.4.2.1 Introduction

The ClassificationScheme DS describes a list of defined terms and their meanings. The ClassificationSchemeBase serves as a abstract base datatype for defining different kinds of classification schemes.

7.4.2.2 ClassificationScheme DS syntax

```
<!-- ##### -->
<!-- Definition of ClassificationScheme DS (7.4.2) -->
<!-- ##### -->

<!-- Definition of ClassificationSchemeBase DS -->
<complexType name="ClassificationSchemeBaseType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Import" type="mpeg7:ReferenceType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="uri" type="anyURI" use="required"/>
      <attribute name="domain" use="optional"/>
      <simpleType>
        <list itemType="mpeg7:xPathAbsoluteSelectorType"/>
      </simpleType>
    </attribute>
  </extension>
</complexContent>
</complexType>

<!-- Definition of ClassificationScheme DS -->
<complexType name="ClassificationSchemeType">
  <complexContent>
    <extension base="mpeg7:ClassificationSchemeBaseType">
      <sequence>
        <element name="Term" type="mpeg7:TermDefinitionType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

7.4.2.3 ClassificationScheme DS semantics

Semantics of the ClassificationSchemeBaseType:

Name	Definition
ClassificationSchemeBaseType	Defines a classification scheme with a set of term definitions (abstract).
Import	Describes a reference to a classification scheme from which to import a set of terms. All terms contained in the imported classification are added into this classification scheme.

Name	Definition
uri	Identifies the classification scheme with a Uniform Resource Identifier (URI). This URI shall uniquely identify this classification scheme. The use of a URN as a classification scheme URI is preferred.
domain	Indicates a set of description elements to which this ClassificationScheme applies. The set is specified using an XPath expression. This attribute is a hint on how the ClassificationScheme can be used in a description. One or more values may be specified. The XPath in this attribute is defined relative to a <i>description</i> root, not the DDL schema. For example: a value of <code>"/Classification/Genre"</code> for domain indicates that the ClassificationScheme can be used for "Genre" elements inside of all "Classification" elements in the description.

Semantics of the ClassificationSchemeType:

Name	Definition
ClassificationSchemeType	Describes a set of terms along with their meaning and relations. ClassificationSchemeType extends ClassificationSchemeBaseType.
Term	Describes a term in the classification scheme. The termID for each term shall be unique within the classification scheme.

7.4.2.4 ClassificationScheme format and importing

7.4.2.4.1 Format of URI for classification schemes

The following defines the normative form of URN for classification schemes defined in this or other parts of ISO/IEC 15938. The URI for all classification schemes defined by ISO/IEC 15938 shall have the following format:

```
urn:mpeg:mpeg7:cs:SchemeName
```

where "urn:mpeg:mpeg7:cs" is a fixed string and identifies the classification schemes as belonging to an ISO/IEC 15938 namespace for classification schemes, irrespective of whether the classification scheme is normative or informative. The *SchemeName* shall be a string made of English letters and numbers and must start with a letter; that is, it shall match the following regular expression:

```
[a-zA-Z][a-zA-Z0-9]*
```

The URN's defined in the MPEG URN namespace are case insensitive and therefore upper and lowercase letters are not distinguished. Additionally, a SchemaName may include version qualifiers in the name, separated from the name by a colon. For example: "MyScheme:2001" attaches the "2001" version qualification to the schema. When they appear, version qualifiers are treated as a part of the whole name of the scheme. Therefore, the name of the scheme "MyScheme:2001" is different from a scheme named "MyScheme."

The use of all URNs in the MPEG namespace, i.e., those URN beginning with a string of the form "urn:mpeg", is reserved for future use. User applications shall not define URN for classification schemes in the MPEG URN namespace. For classification schemes that are not defined in ISO/IEC 15938, the exact form of the URI used to identify and reference a classification scheme is non-normative. However, using a URN is strongly recommended.

7.4.2.4.2 Importing classification schemes

This subclause specifies what it means to import a ClassificationScheme. Importing is used to combine several classification schemes into a single classification scheme.

Let ImportedCS denote the classification scheme being imported and let DefineCS denote the ClassificationScheme being defined. Then the importing of ImportedCS into DefineCS is defined as follows.

- The set of terms defined in ImportedCS is added to the set of terms (TermDefinitions) in DefineCS. If a term with the same termID exists in both ImportedCS and DefineCS, then this is an error.
- For all terms, the definitional information (identifier, name and definition) is incorporated unmodified into DefineCS from ImportedCS
- The set of term relations defined in ImportedCS is added to the set of term relations in DefineCS.

This definition has several consequences as follows:

- An imported term's identifier is the same as its identifier in the original definition. However, when referenced from within a description it shall be referenced by combining the URI for the classification scheme with the term's identifier
- Imported terms may be referenced in relations using the term identifier
- A single term or a term and its immediately related terms shall not be selectively imported. All terms and relations shall be imported.
- When defining and importing classification schemes term identifiers shall be chosen so that that are not likely to conflict.

7.4.2.5 ClassificationScheme DS examples (informative)

7.4.2.5.1 ClassificationScheme DS example: audio domain

The following example shows a classification scheme for the audio domain. The instance of the ClassificationScheme DS is given as follows:

Note: this example is based on a simplified version of the AudioDomainCS classification specified in Annex B.

```
<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>
      </ClassificationScheme>
    </Description>
  </Mpeg7>
```



```

    </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="ja">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:

- URL.** 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 the URN names 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: 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: the definition of term "1.1" is syntactically contained within the definition of term "1". Related terms are syntactically contained within the definition of the term they are being related to. By default, the term relation is "NT*", indicating that the contained is narrower in meaning than its containing term. However, other kinds of relations can be specified by using the "relation" attribute to explicitly indicate the relation between the contained and containing terms.

7.4.2.5.2 ClassificationScheme DS example: genre

The following figure shows an example of a simple classification for genre identified by the URI "urn:example:Genre". In Figure 7, round boxes represent classification schemes and square boxes represent terms.

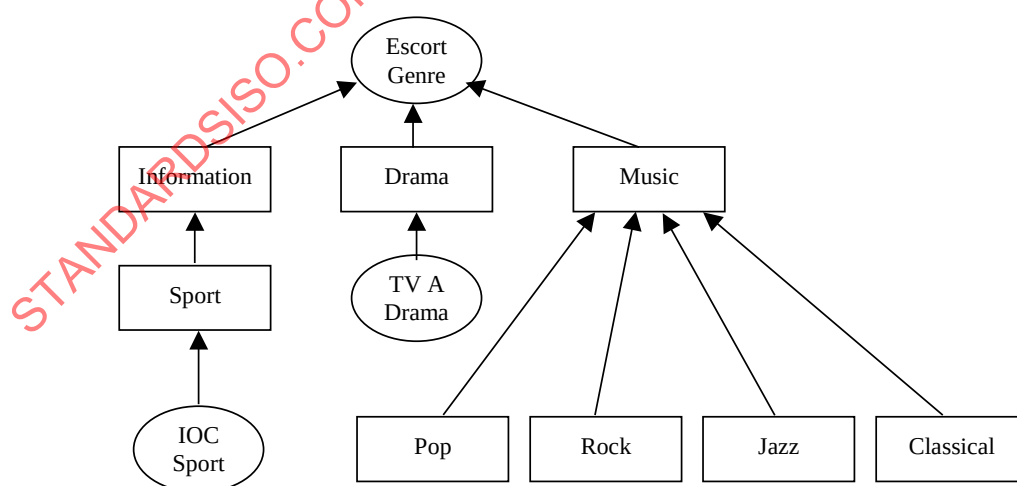


Figure 7 — Illustration of the "Escort2_4:Genre" Genre Classification Scheme

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/" />
      <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>
```

7.4.3 TermDefinition DS

7.4.3.1 Introduction

The TermDefinitionBase DS is an abstract base type for term definitions in classification schemes. It is used as a base for both the TermDefinition DS and the GraphicalTermDefinition DS. The TermDefinition DS describes the meaning of a term. A term definition contains an identifier for a term, a set of names that are human readable, and a textual definition of the term's meaning.

7.4.3.2 TermDefinition DS syntax

```
<!-- ##### -->
<!-- Definition of TermDefinition DS (7.4.3) -->
<!-- ##### -->

<!-- Definition of TermDefinitionBase DS -->
<complexType name="TermDefinitionBaseType" abstract="true">
  <complexContent>
```

```

<extension base="mpeg7:DSType">
  <sequence>
    <element name="Name" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <simpleContent>
          <extension base="mpeg7:TextualType">
            <attribute name="preferred"
              type="boolean" use="optional"/>
          </extension>
        </simpleContent>
      </complexType>
    </element>
    <element name="Definition" type="mpeg7:TextualType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="termID" type="NMTOKEN"/>
</extension>
</complexType>

<!-- Definition of TermDefinition DS -->
<complexType name="TermDefinitionType">
  <complexContent>
    <extension base="mpeg7:TermDefinitionBaseType">
      <sequence>
        <element name="Term" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:TermDefinitionType">
                <attribute name="relation"
                  type="mpeg7:termRelationQualifierType"
                  use="optional" default="NT"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of termRelationQualifier datatype -->
<simpleType name="termRelationQualifierType">
  <union>
    <simpleType>
      <restriction base="NMTOKEN">
        <enumeration value="NT"/> <!-- Narrower term -->
        <enumeration value="BT"/> <!-- Broader term -->
        <enumeration value="RT"/> <!-- Related term -->
        <enumeration value="US"/> <!-- Use term -->
        <enumeration value="UF"/> <!-- Used for term -->
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="mpeg7:termReferenceType"/>
    </simpleType>
  </union>
</simpleType>

```

7.4.3.3 TermDefinition DS semantics

Semantics of the TermDefinitionBaseType:

Name	Definition
TermDefinitionBaseType	Describes a term definition (abstract).
Name	Describes the human readable name for the term. Multiple names can be used for the same term, possibly in different languages. The language of a name is indicated by the <code>xml:lang</code> attribute.
preferred	Indicates whether this is the preferred name for this term. If multiple names exist, only one label per language shall be marked preferred.
Definition	Describes the meaning of the term in natural language prose. Multiple definitions can be used in different languages for the same term. The language of a definition is indicated by the <code>xml:lang</code> attribute.
Term	Describes a term related to the containing term.
relation	Indicates the type of relation existing between the contained item and the containing item. By default, it is "NT", indicating that the contained term is narrower in meaning than this term.
termID	Identifies this term uniquely within a classification scheme.

Semantics of the TermDefinitionType:

Name	Definition
TermDefinitionType	Describes a term definition in a classification scheme. TermDefinitionType extends TermDefinitionBase.
Term	Describes a term related to the containing term.
relation	Indicates the type of the relation existing between the contained item and the containing item. By default, it is "NT", indicating that the contained term is narrower in meaning than this term. Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.

Semantics of the termRelationQualifierType:

Name	Definition
termRelationQualifierType	Describes the type of relation between two controlled terms. In the following, "current term" refers to the containing term and "related term" to the contained term. The following values are defined: BT – Broader term. The related term is more general in meaning than the current term.

Name	Definition
	• NT – Narrower term. The related term is more specific in meaning than the current term. • US – Use instead. The related term is (nearly) synonymous with the current term but the related term is preferred to the current term. • UF – Use for. The related term is (nearly) synonymous with the current term but the current term is preferred to the related term. • RT – Related Term. Related term is not a synonym, quasi-synonym, broader or narrower term, but is associated with the current term.
	Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.

7.4.4 TermUse datatype

7.4.4.1 Introduction

The TermUse datatype allows a term to be used within a description. The TermUse datatype contains a term reference and an optional inline term definition. There are three different ways to use a term in a description:

1. **Term Reference Use.** The term used is taken from a classification scheme. The term is referenced by a term reference (`termReferenceType`), which is either a Uniform Resource Identifier (URI) or term alias (see the subclause defining `termReferenceType`). A term definition (i.e., the Definition element) is not included in an instance of the TermUse datatype in the term reference use.
2. **Inline Term Use.** The term used is taken from a classification scheme and includes a term reference, as in the case of a term reference use. Additionally, it includes the term's information (name, definition, and so forth) directly inside of the TermUse instance. Unlike a free term use, a reference to the term used is mandatory and, unlike a term reference use, the definition information for the term is also included.
3. **Free Term Use.** The term used is described directly inside a TermUse instance. The term is not taken from any classification scheme and therefore has no term reference. This is what distinguishes a free term use from both a term reference use and an inline term use.

7.4.4.2 TermUse datatype syntax

```
<!-- ##### -->
<!-- Definition of TermUse datatype (7.4.4) -->
<!-- ##### -->

<!-- Definition of InlineTermDefinition datatype -->
<complexType name="InlineTermDefinitionType" abstract="true">
  <sequence>
    <element name="Name" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <simpleContent>
          <extension base="mpeg7:TextualType">
            <attribute name="preferred"
              type="boolean" use="optional"/>
          </extension>
        </simpleContent>
      </complexType>
    </element>
  </sequence>
</complexType>
```

```

        </simpleContent>
      </complexType>
    </element>
    <element name="Definition" type="mpeg7:TextualType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="Term" minOccurs="0" maxOccurs="unbounded">
      <complexType>
        <complexContent>
          <extension base="mpeg7:InlineTermDefinitionType">
            <attribute name="relation"
              type="mpeg7:termRelationQualifierType"
              use="optional" default="NT"/>
            <attribute name="termID" type="NMTOKEN"/>
          </extension>
        </complexContent>
      </complexType>
    </element>
  </sequence>
</complexType>

<!-- Definition of TermUse datatype -->
<complexType name="TermUseType">
  <complexContent>
    <extension base="mpeg7:InlineTermDefinitionType">
      <attribute name="href" type="mpeg7:termReferenceType" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

7.4.4.3 TermUse datatype semantics

Semantics of the InlineTermDefinitionType:

Name	Definition
InlineTermDefinitionType	Describes an inline term definition (abstract). This datatype is used as the base for other datatypes that include term definition information.
Name	Describes the human readable name for the term. Multiple names can be used for the same term, possibly in different languages. The language of a name is indicated by the <code>xml:lang</code> attribute.
preferred	Indicates whether this is the preferred name for this term. If multiple names exist, only one label per language shall be marked preferred.
Definition	Describes the definition or meaning of the term in natural language prose. Multiple definitions can be used in different languages for the same term. The language of a definition shall be indicated by the <code>xml:lang</code> attribute.
Term	Describes a term related to the containing term.
relation	Indicates the type of relation existing between the contained item and the containing item. By default, it is "NT", indicating that the contained term is narrower in meaning than this term.
termID	Identifies the contained term that is being related.

Semantics of the TermUseType:

Name	Definition
TermUseType	Describes the use of terms inside a description allowing a term's definition to be inlined. TermUse extends InlineTermDefinitionBaseType
href	Describes a reference to a term using a URI. If href is absent, then the term is defined (its name, definition, or related terms are specified) directly within the TermUse instance using free text. If href is present, then the TermUse is allowed to have only a term reference (i.e. href attribute) without specifying a term definition information. However, as an option, the used term's definition (i.e. the term's name, definition, or related terms) may be described inline. The term's definition in its classification scheme and the inline definition need not have exactly the same value. For example, the names may differ. Differences or conflicts between the classification scheme and inline term definition should be avoided. However, such differences are not considered to be an error. The resolution of such conflicts is not defined by this standard. It is the responsibility of the application using TermUse to try to prevent such mismatches.

7.4.4.4 Validating references to controlled terms (informative)

The check on term references in an instance of TermUse or ControlledTermUse specified in this subclause is optional and need not be performed by a processor. The specification for term reference checking in this subclause shall apply to both the TermUse and ControlledTermUse datatypes.

To resolve the reference to a definition for a term, a processor may use the URI value specified by the href attribute in a TermUse or ControlledTermUse instance. A conforming parser uses that URI to resolve the reference to a resource, whether physical or logical, containing the term's definition. If the term referred to is defined in an ISO/IEC 15938 specified ClassificationScheme, it shall use the "urn:mpeg:mpeg7:cs:schemeName:termID" form specified in the definition of the termReferenceType. Otherwise, the process of resolving the URI is non-normative. Should the URI be resolved to an instance of the TermDefinition DS, it shall occur within the context of the ClassificationScheme DS. A conforming processor shall not signal an error if it cannot successfully resolve the URI used to reference the term.

7.4.4.5 TermUse datatype examples (informative)

7.4.4.5.1 Introduction

In the following subclauses, the three different kinds of TermUse are demonstrated for a genre value. The examples assume that the Genre element is defined to be of type TermUseType.

7.4.4.5.2 TermUse reference examples

The following examples shows the use of the TermUse datatype using the *term reference* mechanism for referring to terms. In these examples, the href attribute includes the references to the term. The following example shows the referencing of a URN: "urn:mpeg:mpeg7:cs:GenreCS:alternativeJazz". Referencing terms in a classification scheme using URNs is used for referring to terms defined in this standard. However, terms defined in classification schemes 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.

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

The following example shows the use of classification schemes aliases, which provide an abbreviated syntax for referring to terms. In this example, 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>.

```
<!-- 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"/>
```

7.4.4.5.3 TermUse inline examples

The following example shows the use of the TermUse datatype using the *inline* mechanism for referring to terms. 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).

```
<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>
```

7.4.4.5.4 TermUse free term examples

The following examples show the use of the TermUse datatype using the *free term use* mechanism for referring to terms. 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.

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

7.4.5 ControlledTermUse datatype

7.4.5.1 Introduction

The ControlledTermUse DS datatype is the *controlled* use of a term from some classification scheme. Unlike the TermUse datatype, which allows a term to be either referenced or written in free text, the ControlledTermUse controls the use of terms by *requiring* a reference to a term.

7.4.5.2 ControlledTermUse datatype syntax

```
<!-- ##### -->
<!-- Definition of ControlledTermUse datatype (7.4.5) -->
<!-- ##### -->

<!-- Definition of ControlledTermUse datatype -->
<complexType name="ControlledTermUseType">
  <complexContent>
    <extension base="mpeg7:InlineTermDefinitionType">
      <attribute name="href" type="mpeg7:termReferenceType" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

7.4.5.3 ControlledTermUse datatype semantics

Semantics of the ControlledTermUseType:

Name	Definition
ControlledTermUseType	Describes the controlled use of terms inside a description, allowing for inclusion of a term's definition inline. ControlledTermUse extends InlineTermDefinitionType.
href	Describes a reference to the term using a URI. The term's definition in the classification scheme and the inline definition need not match. For example, the names may differ. Such mismatches can occur and are not considered an error. It is the responsibility of the application using ControlledTermUse to check for mismatches.

7.4.5.4 ControlledTermUse datatype examples (informative)

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 Institute. 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 as follows:

- **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).

```
<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>
```

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). 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.

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

7.4.6 termReference datatype

7.4.6.1 Introduction

The termReference datatype describes a reference to a term. It supports two forms for term referencing: (1) a URI reference and (2) an abbreviated form for referencing terms in classification schemes that have been assigned aliases.

7.4.6.2 termReference datatype syntax

```
<!-- ##### -->
<!-- Definition of termReferenceType datatype (7.4.6) -->
<!-- ##### -->

<!-- Definition of termReference datatype -->
<simpleType name="termReferenceType">
  <union>
    <simpleType>
      <restriction base="NMTOKEN">
        <pattern value=":[^:]+:[^:]+"/>
        <whiteSpace value="collapse"/>
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="anyURI"/>
    </simpleType>
  </union>
</simpleType>
```

7.4.6.3 termReference datatype semantics

Semantics of the termReferenceType:

Name	Definition
termReferenceType	Describes a reference to a term. The termReferenceType supports two forms for referencing terms: "aliased term references" and URIs. Whether a termReference is determined an alias or URI is determined by whether it satisfies the regular expression given in the pattern facet above. Aliased term references are used to reference terms in classification schemes that have been assigned an alias using a ClassificationSchemeAlias. These references have the lexical form ":[aliasname:termid". Note: this form is not a valid URI, which allows the use of aliases to be syntactically distinguished from a full URI. An alias must be resolved to a classification scheme alias and converted into a URI for the term reference as follows: Resolve alias. When a term reference is an alias, a corresponding instance of ClassificationSchemeAlias having a value of its

Name	Definition
	alias attribute exactly equal to that specified in the termReference shall exist in the description. The alias is said to be <i>resolved</i> to that ClassificationSchemeAlias instance. • Compute URI. The value of the termReference URI is defined to be the value of the href attribute of the resolved ClassificationSchemeAlias instance concatenated with the string ":termid" from the termReferenceValue. • The aliased termReference is then treated as if the URI obtained in the previous step had been used as the value of the termReference. If the termReference's value is a URI, then it shall be a syntactically valid URI.

7.4.6.4 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 7.4. 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.

7.4.7 ClassificationSchemeAlias datatype

7.4.7.1 Introduction

The ClassificationSchemeAlias datatype assigns an alias to a classification scheme so that term references (termReference) can use an abbreviated reference form. The use of classification scheme aliases is specified in the subclause defining the termReference datatype.

7.4.7.2 ClassificationSchemeAlias datatype syntax

```
<!-- ##### -->
<!-- Definition of ClassificationSchemeAlias datatype (7.4.7) -->
<!-- ##### -->

<!-- Definition of ClassificationSchemeAlias datatype -->
<complexType name="ClassificationSchemeAliasType">
  <complexContent>
    <extension base="mpeg7:HeaderType">
      <attribute name="alias" type="NMTOKEN" use="required"/>
      <attribute name="href" type="anyURI" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

```
</complexContent>
</complexType>
```

7.4.7.3 ClassificationSchemeAlias datatype semantics

Semantics of the ClassificationSchemeAliasType:

Name	Definition
ClassificationSchemeAliasType	Describes an alias for a ClassificationScheme referenced by a URI.
alias	Describes the alias assigned to the classification scheme. The scope of the alias assigned shall be the entire description regardless of where the ClassificationSchemeAlias appears in the description.
href	Describes a reference to the classification scheme that is being aliased using a URI. The classification schemes defined in this standard, whether normative or informative, shall be referenced using the identifying value defined by the uri attribute of the ClassificationScheme for that classification scheme.

7.4.8 GraphicalClassificationScheme DS

7.4.8.1 Introduction

The GraphicalClassificationScheme DS allow classification schemes to be used with graphs. A GraphicalClassificationScheme DS contains a list of GraphicalTermDefinitions, each of which is an instance of the Graph DS or a rule for changing one graph into another. A process of changing or rewriting graphs according to rules is called graph production, and the individual rules are called production rules.

Some examples of productions are: changing a node in a graph to a subgraph, changing a relation in a graph to a subgraph, deleting or adding nodes and edges, or gluing two graphs together. These can be used to enrich a description, to modify a description that is no longer accurate, or to join together several descriptions into a single descriptive whole. As with any rewriting system, production rules may be used to represent inference rules.

There are several uses for graphical classification schemes: providing structural templates instantiated by description schemes, validating descriptions having application-defined structure, and as a list of allowable rules for graph productions.

7.4.8.2 GraphicalClassificationScheme DS syntax

```
<!-- ##### -->
<!-- Definition of GraphicalClassificationScheme DS (7.4.8) -->
<!-- ##### -->

<!-- Definition of GraphicalClassificationScheme DS -->
<complexType name="GraphicalClassificationSchemeType">
  <complexContent>
    <extension base="mpeg7:ClassificationSchemeBaseType">
      <sequence>
        <element name="GraphicalTerm" type="mpeg7:GraphicalTermDefinitionType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="strict" type="boolean" use="optional" default="false"/>
    </extension>
```

```
</complexContent>
</complexType>
```

7.4.8.3 GraphicalClassificationScheme DS semantics

Semantics of the GraphicalClassificationSchemeType:

Name	Definition
GraphicalClassificationSchemeType	Describes a classification scheme in which the definition entries are instances of the Graph DS or of one of the GraphRuleDefinitions.
GraphicalTerm	Describes one term in this classification scheme. The termID for each term shall be unique within this classification scheme.
strict	Indicates whether any morphism to a graph in this graphical classification scheme shall be strict. If this attribute has a value of true, only the "urn:cs:mpeg:mpeg7:cs:GraphRelationCS:2001:identity" relation (specified in 7.7.5) can occur in the morphism graph. This attribute is used to restrict use to exact replicas.

7.4.9 GraphicalTermDefinition DS

7.4.9.1 Introduction

The GraphicalTermDefinition DS defines a graphical term in a GraphicalClassificationScheme DS. It derives by extension from TermDefinitionBaseType.

Within a GraphicalTermDefinition, there is one of the following:

1. A Template graph: Describes a model or template for a graph, or to control the list of allowable graphs in a particular application.
2. An Alphabet graph: Defines the structural vocabulary, the "alphabet", used by a set of production rules of the pullback or double pullback type.
3. A GraphRuleDefinition: Defines a graph production rule using one of four paradigms:
 - a. A *pullback* rule is used principally for expanding a graph node into a subgraph.
 - b. A *double pullback* rule is used to replace a subgraph by first collapsing it to a node, then expanding that node to a new subgraph.
 - c. A *pushout* rule is used for gluing together two graphs, and then, optionally deleting some edges and nodes.
 - d. A *double pushout* rule is used to delete some portions of a graph (usually edges), and then to add some other portions.

In order to establish a link between a graph and a graph in a GraphicalTerm in a GraphicalClassificationScheme, a MorphismGraph is used. This is basically a mapping designated by giving the individual node and edge correspondences. In order to qualify as a morphism, for any edge *e* that is mapped to an edge *h*, the end points (source and target) of *e* must be mapped to the end points of *h*.

7.4.9.2 GraphicalTermDefinition DS syntax

```

<!-- ##### -->
<!-- Definition of GraphicalTermDefinition DS (7.4.9) -->
<!-- ##### -->

<!-- Definition of GraphicalTermDefinition DS -->
<complexType name="GraphicalTermDefinitionType">
  <complexContent>
    <extension base="mpeg7:TermDefinitionBaseType">
      <choice>
        <element name="TemplateGraph" type="mpeg7:GraphType"/>
        <element name="AlphabetGraph" type="mpeg7:GraphType"/>
        <element name="ProductionRule"
          type="mpeg7:GraphicalRuleDefinitionType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

7.4.9.3 GraphicalTermDefinition DS semantics

Semantics of the GraphicalTermDefinitionType:

Name	Definition
GraphicalTermDefinitionType	Describes a graphical term definition in a graphical classification scheme. It can contain an instance of a template graph, an alphabet graph or a rule for changing one graph into another.
TemplateGraph	Describes a graph that is used to validate graphs against the classification scheme, or as a template for creating new graphs.
AlphabetGraph	Describes an alphabet graph that is used to define the structural vocabulary, or "alphabet", used by a set of production rules of the pullback or double pullback types.
ProductionRule	Describes a production rule as a term in a graphical classification Scheme. It can be any of four types: pullback, double pullback, pushout, or double pushout.

7.4.10 GraphicalRuleDefinition DS

7.4.10.1 Introduction

The GraphicalRuleDefinition description scheme is an abstract datatype for representing a production rule in a graphical term definition. It is a base type from which the different kind of production rules are derived.

7.4.10.2 GraphicalRuleDefinition DS syntax

```

<!-- ##### -->
<!-- Definition of GraphicalRuleDefinition DS (7.4.10) -->
<!-- ##### -->

<!-- Definition of GraphicalRuleDefinition DS -->

```

```

<complexType name="GraphicalRuleDefinitionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="TemplateGraph" type="mpeg7:GraphType" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

7.4.10.3 GraphicalRuleDefinition DS semantics

Semantics of the GraphicalRuleDefinitionType:

Name	Definition
GraphicalRuleDefinitionType	Describes production rules in graphical classification schemes (abstract).
TemplateGraph	Describes a template graph that an input graph (or subgraph of the input graph) to the production rule must match for the rule to be applicable. If the graph does not match, then the rule is not applicable to that graph. A pair of graphs or subgraphs matches if there is a graph isomorphism between them.

7.4.11 PullbackDefinition DS

7.4.11.1 Introduction

The PullbackDefinition DS is derived from the GraphicalRuleDefinition and defines a pullback production rule. A pullback production rule works as follows:

- The rule is defined by defining a Rule graph, an Alphabet graph, and a morphism between them. The Alphabet graph has sections that correspond to three parts: (1) Context, (2) Interface, and (3) Unknown. These are defined as follows and are illustrated in Figure 8:
 - The *Context* portion of the Alphabet graph corresponds to all the parts of the input graph that shall not be changed, and do not connect to anything that shall be changed. It is usually denoted by a single node, which corresponds to any node type, and that node is connected to itself by a single edge, which corresponds to any edge type. The Context graph may be restricted to limit the form of the context for application of the rule.
 - The *Interface* portion of the Alphabet graph consists of the nodes that shall remain unchanged that are connected to the Unknown portion (which shall change). The Context and the Interface portions are usually connected by edges corresponding to any type of edge, and the nodes in the interface may also correspond to nodes of any type. Once again, this may be restricted if the interface nodes need to be of a specific type, or the connections to the Context need to be constrained.
 - The *Unknown* portion of the Alphabet graph corresponds to the portion of the input graph that shall be rewritten or replaced. It is usually denoted by a single node that corresponds to any node, and a single edge that starts and ends at the single node, also corresponding to any edge; but again, these may be restricted in some applications.
- The Rule graph is also divided into Context, Interface, and Unknown, and the morphism between the Rule graph and the Alphabet graph reflects this: the morphism maps the Rule graph Context to the Alphabet graph Context, the Rule graph Interface to the Alphabet graph Interface, and the Rule graph Unknown to the Alphabet graph Unknown. Usually, the Rule graph Context and Interface portions are identical to the Alphabet graph Context and Interface portions, and the mapping on these subgraphs is the identity.

- When a Morphism graph is provided mapping the input to the Alphabet graph, the part of the input graph mapped to the Context is the Context portion of the input graph. Likewise the parts mapped to the Interface and Unknown portions are the Interface and Unknown portions respectively. Usually, the Pullback rule is used for node replacement. In this case, the node in the input graph that is to be replaced is treated as if it were a node with an edge to itself, making the mapping to the Unknown in the Alphabet graph an identity or isomorphism. However, it need not be, and some techniques map multiple such nodes to the Unknown.
- The product now proceeds as follows:
 - a. For each node on the Alphabet graph, each input graph node mapped to it and each rule graph node mapped to it create one output graph node.
 - b. For each edge, each input graph edge and each rule graph edge create one output graph edge.

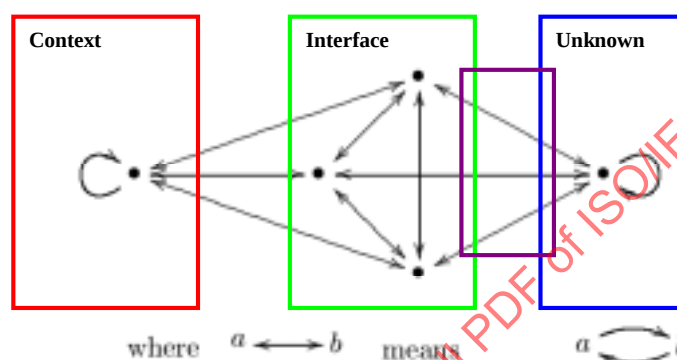


Figure 8 — Relationship of the context, interface, and unknown portions of an Alphabet graph

For example, suppose that there are two nodes in the Alphabet connected by two edges. Suppose as well that the input graph maps two nodes connected by two edges to the two nodes and one of the edges. Finally, suppose that the Rule maps two nodes to the two nodes, and two edges to the same edge. The resultant graph should have two nodes and four edges.

If

- i. the Rule and the Alphabet graph correspond exactly on the Context and Interface nodes,
- ii. and the Context node is a single node with a single edge to itself,
- iii. and if the mapping of the Context and Interface parts of the input graph onto the Alphabet graph are 1-1,

then:

- i. the Context and the Interface nodes, together with their mutual connections shall be mapped faithfully to the output graph, and,
- ii. the Unknown portion of the Rule graph shall be reproduced in place of each node of the input graph which is mapped to the Unknown, and,
- iii. the connecting edges shall contain one edge for each pair of edges, one from the input graph, and one from the Rule graph, which are mapped onto the same edge of the Alphabet graph.

7.4.11.2 PullbackDefinition DS semantics

```

<!-- ##### -->
<!-- Definition of PullbackDefinition DS (7.4.11) -->
<!-- ##### -->

<!-- Definition of PullbackDefinition DS -->
<complexType name="PullbackDefinitionType">
  <complexContent>
    <extension base="mpeg7:GraphicalRuleDefinitionType">
      <sequence>
        <element name="RuleGraph" type="mpeg7:GraphType"/>
        <choice>
          <element name="AlphabetGraph" type="mpeg7:GraphType"/>
          <element name="AlphabetGraphRef"
            type="mpeg7:ControlledTermUseType"/>
        </choice>
        <element name="MorphismGraph" type="mpeg7:MorphismGraphType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

7.4.11.3 GraphicalRuleDefinition DS semantics

Semantics of the PullbackDefinitionType:

Name	Definition
PullbackDefinitionType	Describes a graphical rule definition for a pullback production rule.
RuleGraph	Describes the rule graph for the pullback. This specifies material to be substituted for the unknown portion of the input.
AlphabetGraph	Describes the alphabet graph for the pullback. This specifies how the input is mapped to context, interface, and unknown parts of the alphabet graph.
AlphabetGraphRef	Describes a reference to a term defining an alphabet graph. This allows the same alphabet graph to be used for multiple production rules.
MorphismGraph	Describes the morphism between the rule graph and the alphabet graph, which specifies how to apply the pullback

7.4.12 DoublePullbackDefinition DS

7.4.12.1 Introduction

The DoublePullbackDefinition is derived from GraphicalRuleDefinition and defines a production rule executed by the double pullback production method. Each half of a double pullout is a pullback (see PullbackDefinition DS), but the first pullback is run in reverse. It is usually used to replace a subgraph with a different subgraph. The process is as follows:

- 1) A double pullback is defined by two Rule graphs, each of which is mapped by Morphism graphs to the same Alphabet graph. As in the usual construction for a pullback rule, the Context and Interface portions of each rule graph maps by identity to the Context and Interface portions of the Alphabet. The Unknown portion of the Alphabet graph is usually a unit (a single node with an edge to itself), but need not be for specialized applications.

- 2) The input graph is mapped by a Morphism graph to one of the two rule graphs—normally, the input graph shall "match" only one of the rule graphs and this shall determine the "direction" of the rule. As in the pullback method, the Context and Interface portions of the input graph are defined by the fact that the Morphism graph takes them to the Context and Interface portions of the Rule graph. The Unknown portion of the input graph maps to the Unknown portion of the Rule graph. This implies an intermediary graph, with the Context and Interface portions such that when multiplied (as above) by the Rule graph, one gets the input Context and Interface portions. Likewise, the Unknown portion of the intermediary graph is a factorization of the input graph by the Unknown portion of the Rule graph.
- 3) This intermediary graph becomes the input graph to a second Pullback operation defined by the Alphabet graph and the second Rule graph. The result is the result of this Pullback.
- 4) In the simplest case, the Unknown portion of the input graph maps isomorphically onto the Unknown portion of the Rule graph. Since the factor that would produce this situation is the unit, the Unknown portion of the input is replaced in the intermediary graph by a unit, essentially that part of the input has been collapsed to a node. Then this unit, when fed into the second pullback, is replaced by the Unknown portion of the second Rule graph. Consequently, in this simple case, the operation replaces a section of a graph with something else, and hooks up all the appropriate edges.

7.4.12.2 DoublePullbackDefinition DS syntax

```

<!-- ##### -->
<!-- Definition of DoublePullbackDefinition DS (7.4.12) -->
<!-- ##### -->

<!-- Definition of DoublePullbackDefinition DS -->
<complexType name="DoublePullbackDefinitionType">
  <complexContent>
    <extension base="mpeg7:GraphicalRuleDefinitionType">
      <sequence>
        <element name="RuleGraph" type="mpeg7:GraphType"
          minOccurs="2" maxOccurs="2"/>
        <choice>
          <element name="AlphabetGraph" type="mpeg7:GraphType"/>
          <element name="AlphabetGraphRef"
            type="mpeg7:ControlledTermUseType"/>
        </choice>
        <element name="MorphismGraph" type="mpeg7:MorphismGraphType"
          minOccurs="2" maxOccurs="2"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

7.4.12.3 DoublePullbackDefinition DS semantics

Semantics of the DoublePullbackDefinitionType:

Name	Definition
DoublePullbackDefinitionType	Describes a graphical rule definition for a double pullback production rule.
RuleGraph	Describes the two rule graphs required to implement the double pullback production rule. The first can specify the portion of the input to be collapsed to a node, and the second the replacement graph for the collapsed node.
AlphabetGraph	Describes the alphabet graph for the double pullback production rule. The two rule graphs map to each other through this graph, which serves as an intermediary in the production process.
AlphabetGraphRef	Describes a reference to an alphabet graph. This allows the same alphabet graph to be used for multiple production rules.
MorphismGraph	Describes two morphism graphs, each of which maps a rule graph to an alphabet graph and thus defines the double pullback production rule.

7.4.13 PushoutDefinition DS

7.4.13.1 Introduction

The PushoutDefinition DS derives from GraphicalRuleDefinition DS and defines a rule executed by the pushout production method. Figure 9 illustrates the construction of a pushout rule. There are three definitions of a pushout production method, all of which are accommodated by the PushoutDefinition DS.

1. A pushout operation is a gluing operation. An injective map that maps every node and edge in the left hand side graph (LHS) to an edge or node in the right hand side (RHS) graph is used to form a gluing rule. The image (i.e. the result of applying the mapping) of the LHS graph in the RHS graph constitutes a portion of the RHS graph to be glued, and the rule is applied by mapping the LHS graph to an input graph. For each node in the LHS graph, there is an image node in the RHS, and an image node in the LHS graph, and these two nodes are glued together, resulting in a single node in the output graph. For each edge, the same process is done. This is the basic pushout operation.
2. A pushout can define a partial morphism (such a rule is usually called a single pushout or SPO). This means that a mapping is defined from only a portion of the LHS to the RHS (unlike the first case, where the mapping is total for the LHS graph). When the mapping is defined from the LHS to the input graph, it is interpreted as follows: For every node and edge in the input graph that is the image of a node or edge in the partial morphism, the operation is identical to the basic pushout, and is a gluing. Any any nodes or edges which are the image of portions of the LHS, which are not parts of the partial morphism, are deleted from the final graph. So this production method is a deletion plus gluing operation.
3. There is a pushout method called a 3/2 pushout, which is used for some types of blending in mental spaces. The 3/2 pushout requires that an order (in terms of preferences) be defined for the edges in a set of, and that for any pair of edges there exist a common maximal edge. In the order over edges, identification (equivalence) is considered to be maximal for any edge. A 3/2 pushout is a gluing operation and a map from the LHS to the output graph which is preferred to either of the two composite maps in the pushout, and if, given any other pair of maps from the input and the RHS to a third graph G (which factors into maps to the output and a map from the composite to G) the set of maps from the output to G has a maximal element. For example, if the LHS is linked with the media (or a Segment), and the RHS and input graph are both descriptions of parts of the LHS, then the pushout of these is the "best" description of the media that can be formed by gluing them together.

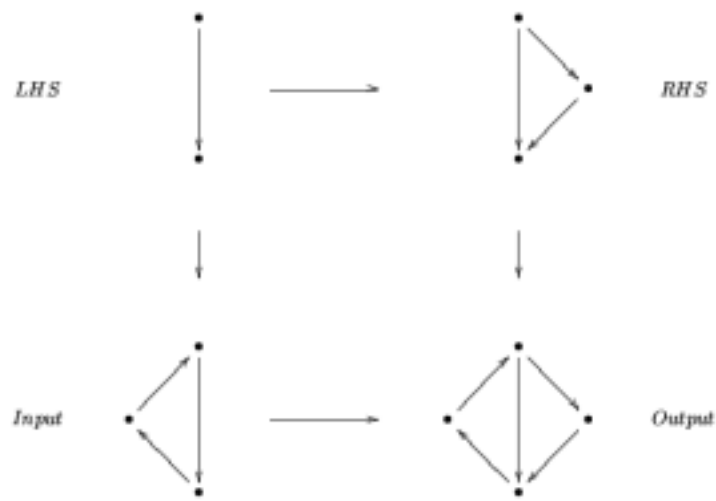


Figure 9 — Illustration of the construction of a pushout rule

These three rules are expressed with the PushoutDefinition DS. If the morphism is total, then a basic pushout is defined, if it is partial, a single pushout is defined. The 3/2 pushout can be implemented by using SemanticState and the MembershipFunctions.

7.4.13.2 PushoutDefinition DS syntax

```
<!-- ##### -->
<!-- Definition of PushoutDefinition DS (7.4.13) -->
<!-- ##### -->

<!-- Definition of PushoutDefinition DS -->
<complexType name="PushoutDefinitionType">
  <complexContent>
    <extension base="mpeg7:GraphicalRuleDefinitionType">
      <sequence>
        <element name="LHSGraph" type="mpeg7:GraphType"/>
        <element name="RHSGraph" type="mpeg7:GraphType"/>
        <element name="MorphismGraph" type="mpeg7:MorphismGraphType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

7.4.13.3 PushoutDefinition DS semantics

Semantics of the PushoutDefinitionType:

Name	Definition
PushoutDefinitionType	Describes a graphical rule definition for pushout, or single pushout, production rules.
LHSGraph	Describe the left hand side (LHS) of the pushout rule.

Name	Definition
RHSGraph	Describes the right-hand-side of the pushout rule.
MorphismGraph	Describes a morphism between the LHSGraph and the RHSGraph. If this morphism is partial (I.e. a mapping in which not all edges and nodes in the LHSGraph are mapped) then the rule is a single pushout rule, and the unreferenced portion of the LHSGraph shall be deleted. If the morphism is total, then the operation is a simple gluing operation.

7.4.14 DoublePushoutDefinition DS

7.4.14.1 Introduciton

The DoublePushoutDefinition DS defines a rule executed by the double pushout (DPO) production method. DoublePushoutDefinition DS extends GraphicalRuleDefinition DS. This production rule is a cut and paste method, usually used with edge replacement. Figure 10 illustrates a double pushout rule. It is specified as follows:

- 1) The rule is specified by three graphs, connected by two morphisms. The morphisms specify injections (embeddings) of the Context graph into a LHS graph and a RHSGraph. The Context graph corresponds to that portion of the original graph that shall remain unchanged. The portion of the LHS graph that is not in the image of the Context graph corresponds to the portion to be deleted, and the portion of the RHS graph that is not in the image of the Context graph corresponds to what shall be added.
- 2) An application of the rule is made by specifying a morphism (using a MorphismGraph) between the LHSGraph and an input graph. There are rules for this, known as *dangling conditions*:
 - a. No point in the input graph can be the image of both a deleted portion and an unchanged portion of the LHSGraph.
 - b. No edges can be left dangling (unconnected because an endpoint was deleted) by the morphism.
- 3) A graph is created that corresponds to the ContextGraph, as the graph that would need to be glued to the LHSGraph to form the input graph. Then this graph and the RHSGraph are glued along the images of the ContextGraph to form the output. These two gluing operations give the pushout operations its name.

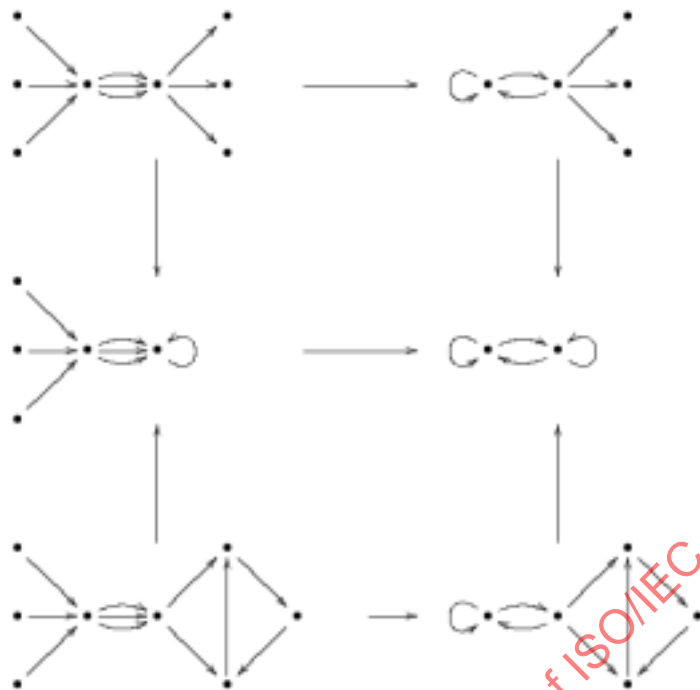


Figure 10 — Illustration of a double pushout rule

7.4.14.2 DoublePushoutDefinition DS syntax

```
<!-- ##### -->
<!-- Definition of DoublePushoutDefinition DS (7.4.14) -->
<!-- ##### -->

<!-- Definition of DoublePushoutDefinition DS -->
<complexType name="DoublePushoutDefinitionType">
  <complexContent>
    <extension base="mpeg7:GraphicalRuleDefinitionType">
      <sequence>
        <element name="LHSGraph" type="mpeg7:GraphType"/>
        <element name="ContextGraph" type="mpeg7:GraphType"/>
        <element name="RHSGraph" type="mpeg7:GraphType"/>
        <element name="MorphismGraph" type="mpeg7:MorphismGraphType"
          minOccurs="2" maxOccurs="2"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

7.4.14.3 DoublePushoutDefinition DS semantics

Semantics of the DoublePushoutDefinitionType:

Name	Definition
DoublePushoutDefinitionType	Describes a graphical rule definition for double pushout production rules.

Name	Definition
LHSGraph	Describe a graph specifying the left-hand-side of the pushout rule.
ContextGraph	Describe a graph specifying the context graph of the double pushout operation.
RHSGraph	Describes a graph specifying the right hand side of the pushout rule.
MorphismGraph	Describes two morphism graphs, one from the context to the LHS graph and one from the context to the RHS graph needed to specify the double pushout production rule.

7.4.15 GraphicalTermDefinition DS examples (informative)

7.4.15.1 Introduction

The following subclause presents informative examples showing how each of the four different kinds of graph productions can be expressed using the GraphicalTermDefinition DS.

7.4.15.2 Pullback example

Figure 11 shows an example of a pullback operation where the input graph a, b, c, d, e, f is modified by the rule graph 1, 2, 3, 4, 5. This example is presented using a category theoretic diagram of a pullback graph production:

- The diagram shows the input graph being mapped to the alphabet graph by the morphism F and the rule graph in the upper right being mapped to the alphabet graph by the morphism G . The output graph, shown in the top left, is formed by "pulling back" the two mappings to form the output graph.
- The output of the rule on the input graph can be seen as the result of the replacing node b by nodes 2, 3 and edge d by edges 2, 3.
- The morphism F maps the input graph to the alphabet graph. It maps nodes a of the input graph to node x of the alphabet graph, nodes b and c to nodes y , edges d and f map to z and edge e to w in the alphabet graph. This morphism F is also called an occurrence.
- The morphism G maps the rule graph to the alphabet graph. It maps nodes 2, 3 of the rule graph to node y of the alphabet graph and node 1 to x . This morphism expresses an operation of graph transformation together with an embedding relation.
- The application of the rule G to the occurrence F is obtained by computing the pullback of the two graph morphisms F and G . Technically, the pullback is a subgraph of the product of 1, 2, 3, 4, 5 by a, b, c, d, e, f with respect to the mappings G and F .

In this example, nodes c and b are "multiplied" by nodes 2 and 3 as they map to the same node in the alphabet graph, as do nodes a and 1 and so on, to form the output graph. The construction of the resulting edges is similar for the morphisms on the edges. The edge d maps to a part of the alphabet graph that has no counterpart in the rule graph, so it does not end up in the output graph. The output graph contains a copy of the rule graph everywhere that there is a counterpart in the input.

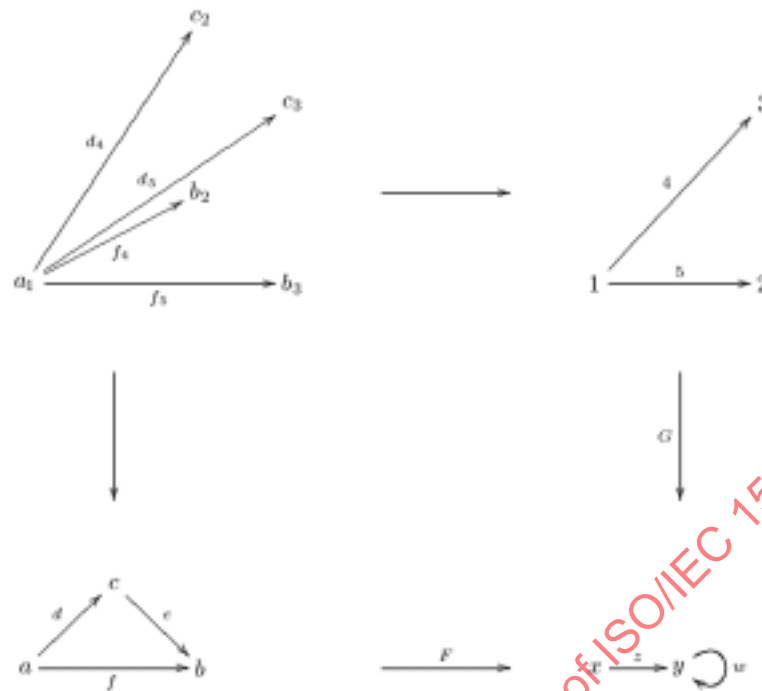


Figure 11 — Illustration of a pullback operation on an input graph modified by a rule graph

The mapping F in the above figure can be written as an adjacency matrix with a, b, c, d, e, f as rows and x, y, z, w as columns. In this matrix, wherever there is a mapping (e.g. from the node a to node x) there is a one. The G mapping can be represented similarly. $\{a, b, c, d, e, f\}$ is the input graph, and the result has a node or edge for every 1 in the matrix: In this matrix, $\{x, y, z, w\}$ form the rows, and $\{1, 2, 3, 4, 5\}$ form the columns.

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad G' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Finally, if the matrix representing the morphism F is multiplied by the transpose of the matrix representing the morphism G , the output of the pullout operation is given as follows, where the rows, x, y, z, w , represent the alphabet graph and the columns, $1, 2, 3, 4, 5$, represent the rule graph:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

This illustrates a simple algorithm for calculating the result of a pullback operation: Create the matrix representing each mapping or morphism graph and compute their matrix product. Interpret the resulting matrix as a list of the possible combinations of input and rule elements.

A pullout operation can perform node replacement, which is the easiest operation done by pullback. Divide the input graph is divided up into context, boundary and unknown parts. If the unknown part is mapped to a single node (like the node *y* above), and if the rule graph duplicates the alphabet graph except for what is mapped to this node, then the output shall be the result of replacing the unknown part of the input graph with the unknown part of the rule graph. The result is *node replacement*. When rules are constructed in this three part form, the necessary matrix representations of the functions need only specify the map to the boundary and unknown parts of the alphabet graph.

Consider a production rule that rewrites (or infers from) a semantic description so that objects with "arms" also have "hands." Then, "hands" are added to the nodes using the following rule:

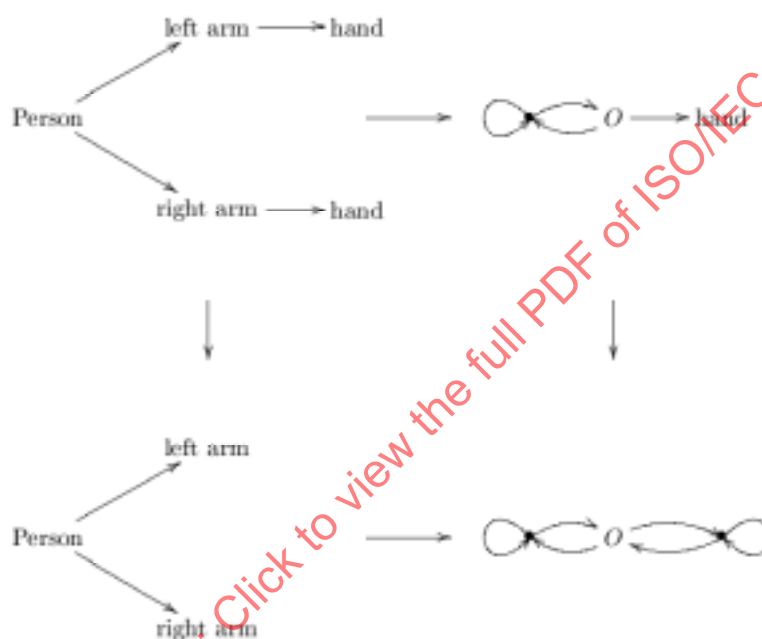


Figure 12 — Example graphical production rule represented by Graphical Classification Scheme

This example consist of three parts:

(1) Graphical Classification DS. The graphical classification scheme contains two terms:

- a. The Alphabet graph term (term "1"). The alphabet graph contains three nodes representing the unknown, boundary, and context. In this graph the boundary node is identified as an Object node.
- b. The Production Rule (term "2"). Inside of the production are:
 - i. The Rule graph. This graph consists of three nodes: a node representing the "hand", a boundary node which is an object, and a third node representing the context.
 - ii. A graph morphism from the rule graph to the alphabet graph. This morphism maps the node representing "hand" to the "unknown" node in the alphabet graph, the boundary and context nodes are mapped correspondingly.

- c. Morphism Graph. This establishes the mappings between the rule graph (referred to by the SourceGraphRef) and the alphabet graph that was defined in term "1" (referred to by TargetGraphTermRef).
- (2) Semantic DS. This instance of the Semantic DS (see clause 12) describes the situation where "person" has a "leftarm" and a "rightarm." This configuration is represented by a semantic graph "persongraph."
- (3) Morphism Graph. The morphism graph (see 7.7.4) represents an application of the production rule defined in term "2" above (SourceGraphRef is "persongraph") to the alphabet graph defined in term 1 (TargetGraphTermRef is "urn:example:cs:Graph1:1").

The complete example description follows:

```
<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationSchemeBase xsi:type="GraphicalClassificationSchemeType"
      uri="urn:example:cs:Graph1">
      <GraphicalTerm termID="1">
        <AlphabetGraph id="ag">
          <Node id="unknown">
            <Name href="urn:example:any"/>
          </Node>
          <Node id="boundary">
            <Name href="urn:example:Object"/>
          </Node>
          <Node id="context">
            <Name href="urn:example:any"/>
          </Node>
          <Relation id="ag1" source="#unknown" target="#unknown"
            type="urn:example:any"/>
          <Relation id="ag2" source="#boundary" target="#unknown"
            type="urn:example:any"/>
          <Relation id="ag3" source="#unknown" target="#boundary"
            type="urn:example:any"/>
          <Relation id="ag4" source="#boundary" target="#context"
            type="urn:example:any"/>
          <Relation id="ag5" source="#context" target="#boundary"
            type="urn:example:any"/>
          <Relation id="ag6" source="#context" target="#context"
            type="urn:example:any"/>
        </AlphabetGraph>
      </GraphicalTerm>
      <GraphicalTerm termID="2">
        <ProductionRule xsi:type="PullbackDefinitionType">
          <RuleGraph id="rg">
            <Node id="rghand">
              <Name href="urn:example:HandObject"/>
            </Node>
            <Node id="rgboundary">
              <Name href="urn:example:Object"/>
            </Node>
            <Node id="rgcontext">
              <Name href="urn:example:any"/>
            </Node>
            <Relation id="rg1" source="#rgboundary" target="#rghand"
              type="urn:example:contains"/>
            <Relation id="rg2" source="#rgboundary" target="#rgcontext"
              type="urn:example:any"/>
            <Relation id="rg3" source="#rgcontext" target="#rgboundary">

```

```

        type="urn:example:any"/>
        <Relation id="rg4" source="#rgcontext" target="#rgcontext"
        type="urn:example:any"/>
    </RuleGraph>
    <AlphabetGraphRef href="urn:example:cs:Graph1:1"/>
    <MorphismGraph>
        <Relation source="#rghand" target="#unknown"
        type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
        <Relation source="#rgboundary" target="#boundary"
        type="urn:example:any"/>
        <Relation source="#rgcontext" target="#context"
        type="urn:example:any"/>
        <Relation source="#rg1" target="#ag2" type="urn:example:any"/>
        <Relation source="#rg2" target="#ag4" type="urn:example:any"/>
        <Relation source="#rg3" target="#ag5" type="urn:example:any"/>
        <Relation source="#rg4" target="#ag6" type="urn:example:any"/>
        <SourceGraphRef href="#rg"/>
        <TargetGraphTermRef href="urn:example:cs:Graph1:1"/>
    </MorphismGraph>
</ProductionRule>
</GraphicalTerm>
</ClassificationSchemeBase>
</Description>
<Description xsi:type="SemanticDescriptionType">
    <Semantics>
        <Label>
            <Name>Semantic description of "person" with "leftarm" and "rightarm".
            </Name>
        </Label>
        <SemanticBase xsi:type="ObjectType" id="person">
            <Label>
                <Name>Person</Name>
            </Label>
        </SemanticBase>
        <SemanticBase xsi:type="ObjectType" id="leftarm">
            <Label>
                <Name>left arm</Name>
            </Label>
        </SemanticBase>
        <SemanticBase xsi:type="ObjectType" id="rightarm">
            <Label>
                <Name>right arm</Name>
            </Label>
        </SemanticBase>
        <Graph id="persongraph">
            <Relation id="person1" type="urn:example:has" source="#person"
            target="#leftarm"/>
            <Relation id="person2" type="urn:example:has" source="#person"
            target="#rightarm"/>
        </Graph>
    </Semantics>
</Description>
<Description xsi:type="SemanticDescriptionType">
    <Semantics>
        <Label>
            <Name> Morphisms graph of relations </Name>
        </Label>
        <Graph xsi:type="MorphismGraphType">
            <Relation source="#object3" target="#tom1"
            type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
            <Relation source="#object4" target="#mary1"

```

```

        type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
        <Relation source="#lhs1" target="#ew1"
            type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
        <SourceGraphTermRef href="urn:example:cs:Graph2:1"/>
        <TargetGraphRef href="#eastwestgraph"/>
    </Graph>
</Semantics>
</Description>
</Mpeg7>

```

7.4.15.3 Double pullback example

In the double pullback, the input graph is mapped to a rule graph. This is used, usually, to collapse a particular part of the input graph to a node, so it can be replaced. The replacement is the second rule. It gives more control for this type of operation than a pullback rule, since one can validate how the collapsed section should look.

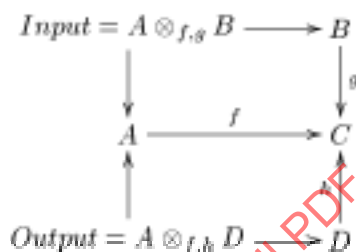


Figure 13 — Illustration of double pullback in which an input graph is mapped to a rule graph

In the following example, this is used to take a beach, described as having Sun, Waves, and Fish, and convert it to a more northerly description. The input is collapsed to Beach, with the necessary placeholder, then the second rule pastes in Clouds, Icebergs, and Seals.

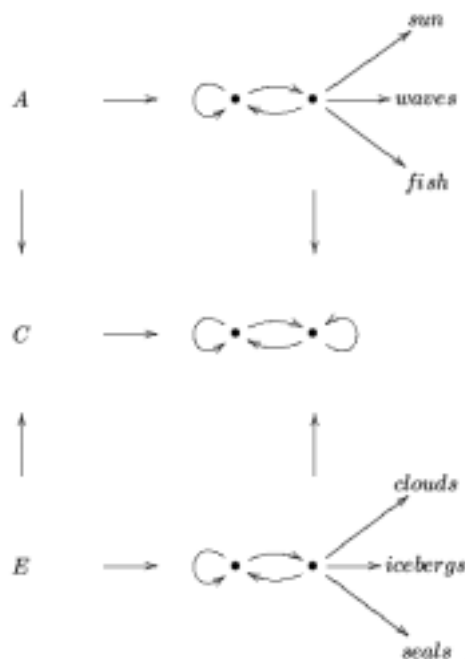


Figure 14 — An example of a double pushout rule

```

<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationSchemeBase xsi:type="GraphicalClassificationSchemeType"
      uri="urn:example:cs:Graph2">
      <GraphicalTerm termID="1">
        <AlphabetGraph id="ag">
          <Node id="unknown">
            <Name href="urn:example:any"/>
          </Node>
          <Node id="context">
            <Name href="urn:example:any"/>
          </Node>
          <Relation id="ag1" source="#unknown" target="#unknown"
            type="urn:example:any"/>
          <Relation id="ag2" source="#unknown" target="#context"
            type="urn:example:any"/>
          <Relation id="ag3" source="#context" target="#unknown"
            type="urn:example:any"/>
          <Relation id="ag4" source="#context" target="#context"
            type="urn:example:any"/>
        </AlphabetGraph>
      </GraphicalTerm>
      <GraphicalTerm termID="2">
        <ProductionRule xsi:type="DoublePullbackDefinitionType">
          <RuleGraph id="rule1">
            <Node id="rg1context">
              <Name href="urn:example:any"/>
            </Node>
            <Node id="sun">
              <Name href="urn:example:Sun"/>
            </Node>
            <Node id="waves">
              <Name href="urn:example:waves"/>
            </Node>
          </RuleGraph>
        </ProductionRule>
      </GraphicalTerm>
    </ClassificationSchemeBase>
  </Description>

```

```

</Node>
<Node id="fish">
  <Name href="urn:example:fish"/>
</Node>
<Relation id="rg11" source="#rg1context" target="#context"
  type="urn:example:any"/>
<Relation id="rg12" source="#rg1context" target="#sun"
  type="urn:example:contains"/>
<Relation id="rg13" source="#rg1context" target="#waves"
  type="urn:example:contains"/>
<Relation id="rg14" source="#rg1context" target="#fish"
  type="urn:example:contains"/>
</RuleGraph>
<RuleGraph id="rule2">
  <Node id="rg2context">
    <Name href="urn:example:any"/>
  </Node>
  <Node id="clouds">
    <Name href="urn:example:clouds"/>
  </Node>
  <Node id="icebergs">
    <Name href="urn:example:icebergs"/>
  </Node>
  <Node id="seals">
    <Name href="urn:example:seals"/>
  </Node>
  <Relation id="rg21" source="#rg2context" target="#context"
    type="urn:example:any"/>
  <Relation id="rg22" source="#rg2context" target="#clouds"
    type="urn:example:contains"/>
  <Relation id="rg23" source="#rg2context" target="#icebergs"
    type="urn:example:contains"/>
  <Relation id="rg24" source="#rg2context" target="#seals"
    type="urn:example:contains"/>
</RuleGraph>
<AlphabetGraphRef href="urn:example:cs:Graph2:1"/>
<MorphismGraph>
  <Relation source="#rg1context" target="#context"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#waves" target="#unknown"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#sun" target="#unknown"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#fish" target="#unknown"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#rg11" target="#ag4"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#rg12" target="#ag3"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#rg13" target="#ag3"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#rg14" target="#ag3"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <SourceGraphRef href="#rule1"/>
  <TargetGraphRef href="#ag"/>
</MorphismGraph>
<MorphismGraph>
  <Relation source="#rg2context" target="#context"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#clouds" target="#unknown"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>

```

```

    <Relation source="#icebergs" target="#unknown"
      type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
    <Relation source="#waves" target="#unknown"
      type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
    <Relation source="#rg21" target="#ag4"
      type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
    <Relation source="#rg22" target="#ag3"
      type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
    <Relation source="#rg23" target="#ag3"
      type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
    <Relation source="#rg24" target="#ag3"
      type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
    <SourceGraphRef href="#rule2"/>
    <TargetGraphRef href="#ag"/>
  </MorphismGraph>
</ProductionRule>
</GraphicalTerm>
</ClassificationSchemeBase>
</Description>
</Mpeg7>

```

7.4.15.4 Pushout example

A pushout rule performs a "gluing" operation. The gluing template is encoded using an AlphabetGraph. If the alphabet graph is mapped to some portion of the input graph, then the pushout equates that part of the input with the part of the rule mapped to the same place. Figure 15 illustrates the pushout process.

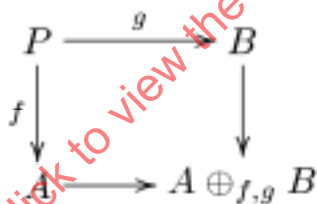


Figure 15 — Illustration of pushout process in which a product graph is used as a gluing template

The following example takes a semantic description of a beach, described as having just beach and sky, and applies a rule that describes a beach as having both a beach and waves. The result (the output of the rule) is a beach with both sky and waves.

```

<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationSchemeBase xsi:type="GraphicalClassificationSchemeType"
      uri="urn:example:cs:Graph3">
      <GraphicalTerm termID="1">
        <ProductionRule xsi:type="PushoutDefinitionType">
          <LHSGraph id="lhs">
            <Node id="agbeach">
              <Name href="urn:example:beach"/>
            </Node>
          </LHSGraph>
          <RHSGraph id="rhs">
            <Node id="rgbeach">
              <Name href="urn:example:beach"/>
            </Node>
            <Node id="rgwaves">

```



```

        <Name href="urn:example:waves"/>
      </Node>
    </RHSGraph>
    <MorphismGraph>
      <Relation source="#agbeach" target="#rgbeach"
        type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
      <SourceGraphRef href="#lhs"/>
      <TargetGraphRef href="#rhs"/>
    </MorphismGraph>
  </ProductionRule>
</GraphicalTerm>
</ClassificationSchemeBase>
</Description>
<Description xsi:type="SemanticDescriptionType">
  <Semantics>
    <Label>
      <Name>Beach scene</Name>
    </Label>
    <SemanticBase xsi:type="ObjectType" id="beach1">
      <Label>
        <Name>Beach</Name>
      </Label>
    </SemanticBase>
    <SemanticBase xsi:type="ObjectType" id="sky1">
      <Label>
        <Name>Sky</Name>
      </Label>
    </SemanticBase>
    <Graph id="beachgraph">
      <Node id="beach" href="#beach1"/>
      <Node id="sky" href="#sky1"/>
    </Graph>
  </Semantics>
</Description>
<Description xsi:type="SemanticDescriptionType">
  <Semantics>
    <Label>
      <Name> Morphisms graph of relations </Name>
    </Label>
    <Graph xsi:type="MorphismGraphType">
      <Relation source="#agbeach" target="#beach"
        type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
      <SourceGraphTermRef href="urn:example:cs:Graph3:1"/>
      <TargetGraphRef href="#beachgraph"/>
    </Graph>
  </Semantics>
</Description>
</Mpeg7>

```

7.4.15.5 Double pushout example

The double pushout technique is most frequently applied to edge rewriting. It is a cut and paste operation in which the left hand side graph of the graph production is cut to produce a center graph, then whatever is needed is pasted to create the right hand graph. This is illustrated in Figure 16. To use a double pushout rule, make a map from the left hand side graph to an input graph. The map implies the rest of the diagram.

In Figure 16 the rule makes the statement "If Tom is to the west of Mary, then Mary is to the east of Tom." In detail, the map m takes $O1$ to Tom, $O2$ to Mary, and the relation west to west. This implies that d should take $O1$ to Tom and $O2$ to Mary in the center mapping. The center mapping d together with r imply that the map m^* should take $O1$ to Tom, $O2$ to Mary, and map the relation east between them. The double pushout operation has some symmetry:

Where it makes sense, instead of making a morphism from the input to the left hand side of the rule, one can make a morphism to the right hand side and get as a result something like the left hand side. There are two caveats: do not leave edges dangling, and don't map any node to both the cut away part and the part that remains (highly unlikely in applications).

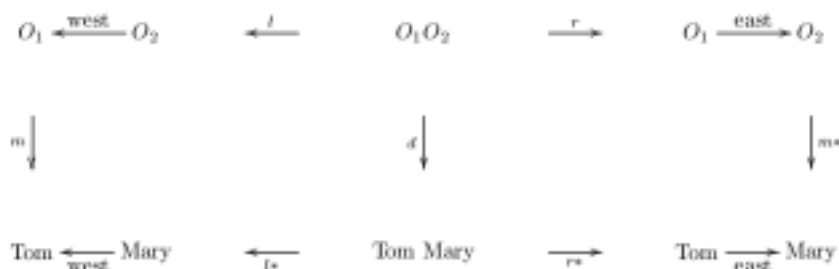


Figure 16 — Illustration of double pushout technique applied to edge rewriting of graphs

7.4.15.6 GraphicalTerm description examples

```
<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationSchemeBase xsi:type="GraphicalClassificationSchemeType"
      uri="urn:example:cs:Graph4">
      <GraphicalTerm termID="1">
        <ProductionRule xsi:type="DoublePushoutDefinitionType">
          <LHSGraph id="LHS">
            <Node id="object3">
              <Name href="urn:example:object"/>
            </Node>
            <Node id="object4">
              <Name href="urn:example:object"/>
            </Node>
            <Relation id="lhs1" type="urn:example:west" source="#object4"
              target="#object3"/>
          </LHSGraph>
          <ContextGraph id="context">
            <Node id="object1">
              <Name href="urn:example:object"/>
            </Node>
            <Node id="object2">
              <Name href="urn:example:object"/>
            </Node>
          </ContextGraph>
          <RHSGraph id="RHS">
            <Node id="object5">
              <Name href="urn:example:object"/>
            </Node>
            <Node id="object6">
              <Name href="urn:example:object"/>
            </Node>
            <Relation id="rhs1" type="urn:example:east" source="#object5"
              target="#object6"/>
          </RHSGraph>
          <MorphismGraph>
            <Relation id="left1"
              type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"
              source="#object1" target="#object3"/>
            <Relation id="left2"
```

```

        type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"
        source="#object2" target="#object4"/>
        <SourceGraphRef href="#context"/>
        <TargetGraphRef href="#LHS"/>
    </MorphismGraph>
    <MorphismGraph>
        <Relation id="rgt1"
            type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"
            source="#object1" target="#object3"/>
        <Relation id="rgt2"
            type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"
            source="#object2" target="#object4"/>
        <SourceGraphRef href="#context"/>
        <TargetGraphRef href="#LHS"/>
    </MorphismGraph>
    </ProductionRule>
</GraphicalTerm>
</ClassificationSchemeBase>
</Description>
<Description xsi:type="SemanticDescriptionType">
    <Semantics>
        <Label>
            <Name>East West</Name>
        </Label>
        <SemanticBase xsi:type="ObjectType" id="tom">
            <Label>
                <Name>Tom</Name>
            </Label>
        </SemanticBase>
        <SemanticBase xsi:type="ObjectType" id="mary">
            <Label>
                <Name>Mary</Name>
            </Label>
        </SemanticBase>
        <Graph id="eastwestgraph">
            <Node id="tom1" href="#tom"/>
            <Node id="mary1" href="#mary"/>
            <Relation id="ew1" type="urn:example:west" source="#Tom"
                target="#Mary"/>
        </Graph>
    </Semantics>
</Description>
<Description xsi:type="SemanticDescriptionType">
    <Semantics>
        <Label>
            <Name> Morphisms graph of relations </Name>
        </Label>
        <Graph xsi:type="MorphismGraphType">
            <Relation source="#object3" target="#tom1"
                type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
            <Relation source="#object4" target="#mary1"
                type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
            <Relation source="#lhs1" target="#ew1"
                type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
            <SourceGraphTermRef href="urn:example:cs:Graph2:1"/>
            <TargetGraphRef href="#eastwestgraph"/>
        </Graph>
    </Semantics>
</Description>
</Mpeg7>

```

7.5 Description of agents

7.5.1 Introduction

This subclause defines tools for describing entities that are agents: persons, organizations, and groups of persons.

7.5.2 Agent DS

7.5.2.1 Introduction

The Agent DS represents the abstract concept of an "agent," which is an entity that is regarded as being responsible for some action, such as the creation of multimedia content. Concretely, an agent can be a person (e.g., actor, director, character, dubbing actor), an organization (e.g., a company) or a group of persons (e.g. a musical ensemble).

7.5.2.2 Agent DS syntax

```
<!-- ##### -->
<!-- Definition of Agent DS (7.5.2) -->
<!-- ##### -->

<!-- Definition of Agent DS -->
<complexType name="AgentType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Icon" type="mpeg7:MediaLocatorType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

7.5.2.3 Agent DS semantics

Semantics of the AgentType:

Name	Definition
AgentType	Describes an agent (abstract). The agent can be a person, a group of persons, or an organization.
Icon	Describes a multimedia "icon" associated with this agent. An icon is taken to be any multimedia content (e.g. image, audio, video, and so forth) that serves as an iconic representation of this agent.

7.5.3 Person DS

7.5.3.1 Introduction

The Person DS describes a person, whether real, historical or fictional. The Person DS can be used to describe an actor, a character, or a producer.

7.5.3.2 Person DS syntax

```

<!-- ##### -->
<!-- Definition of Person DS (7.5.3) -->
<!-- ##### -->

<!-- Definition of Person DS -->
<complexType name="PersonType">
  <complexContent>
    <extension base="mpeg7:AgentType">
      <sequence>
        <choice minOccurs="1" maxOccurs="unbounded">
          <element name="Name" type="mpeg7:PersonNameType"/>
          <element name="NameTerm" type="mpeg7:ControlledTermUseType"/>
        </choice>
        <element name="Affiliation" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <choice>
              <element name="Organization" type="mpeg7:OrganizationType"/>
              <element name="OrganizationRef" type="mpeg7:ReferenceType"/>
              <element name="PersonGroup" type="mpeg7:PersonGroupType"/>
              <element name="PersonGroupRef" type="mpeg7:ReferenceType"/>
            </choice>
          </complexType>
        </element>
        <element name="Citizenship" type="mpeg7:countryCode"
          minOccurs="0" maxOccurs="unbounded"/>
        <choice minOccurs="0">
          <element name="Address" type="mpeg7:PlaceType"/>
          <element name="AddressRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="ElectronicAddress" type="mpeg7:ElectronicAddressType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

7.5.3.3 Person DS semantics

Semantics of the PersonType:

Name	Definition
PersonType	Describes an individual person.
Name	Describes a name associated with this person. Multiple names are allowed. The interpretation of different names is not defined in this standard.
NameTerm	Describes a name associated with this person using a term from a classification scheme.
Affiliation	Describes the organizations or person groups with which this person is affiliated. For example, the company for which the individual works, the sports team on which they place, and so on. A person may be affiliated with more than one organization or group.
Organization	Describes an organization that this person is affiliated with.

Name	Definition
OrganizationRef	Describes a reference to an organization that this person is affiliated with.
PersonGroup	Describes a group that this person is affiliated with.
PersonGroupRef	Describes a reference to a person group that this person is affiliated with.
Citizenship	Describes the countries in which this person is a legal citizen. A person may hold multiple citizenships. The legal definition of citizenship is not defined in this standard and may vary from country to country.
Address	Describes a place that is the address where this person can be located.
AddressRef	Describes a reference to a place that is the address where this person can be located.
ElectronicAddresses	Describes the electronic address information for this person.

7.5.3.4 Person DS examples (informative)

The following example shows the use of Person DS for describing a production assistant who is an individual called "John Smith" in English and "Jean Smith" in French who is affiliated with ISO and whose e-mail address is "john.smith@iso.ch".

```

<Mpeg7>
  <Description xsi:type="CreationDescriptionType">
    <CreationInformation>
      <Creation>
        <Title/>
        <Creator>
          <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:PRODUCTION-ASSISTANT"/>
          <Agent xsi:type="PersonType">
            <Name xml:lang="en">
              <GivenName>John</GivenName>
              <FamilyName>Smith</FamilyName>
            </Name>
            <Name xml:lang="fr">
              <GivenName>Jean</GivenName>
              <FamilyName>Smith</FamilyName>
            </Name>
            <Affiliation>
              <Organization>
                <Name>International Standards Organization</Name>
              </Organization>
            </Affiliation>
            <ElectronicAddress>
              <Email>john.smith@iso.ch</Email>
            </ElectronicAddress>
          </Agent>
        </Creator>
      </Creation>
    </CreationInformation>
  </Description>
</Mpeg7>

```

7.5.4 PersonGroup DS

7.5.4.1 Introduction

The PersonGroup DS describes a group of persons having a collective title and whose member persons are easily identified, i.e. a musical group, a sports team, or a dance ensemble.

7.5.4.2 PersonGroup DS syntax

```
<!-- ##### -->
<!-- Definition of PersonGroup DS (7.5.4) -->
<!-- ##### -->

<!-- Definition of PersonGroup DS -->
<complexType name="PersonGroupType">
  <complexContent>
    <extension base="mpeg7:AgentType">
      <sequence>
        <element name="Name" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="mpeg7:TextualType">
                <attribute name="type" use="optional">
                  <simpleType>
                    <restriction base="NMTOKEN">
                      <enumeration value="former"/>
                      <enumeration value="variant"/>
                      <enumeration value="main"/>
                    </restriction>
                  </simpleType>
                </attribute>
              </extension>
            </simpleContent>
          </complexType>
        </element>
        <element name="NameTerm" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:ControlledTermUseType">
                <attribute name="type" use="optional">
                  <simpleType>
                    <restriction base="NMTOKEN">
                      <enumeration value="former"/>
                      <enumeration value="variant"/>
                      <enumeration value="main"/>
                    </restriction>
                  </simpleType>
                </attribute>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="Kind" type="mpeg7:TermUseType" minOccurs="0"/>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Member" type="mpeg7:PersonType"/>
          <element name="MemberRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0">
          <element name="Jurisdiction" type="mpeg7:PlaceType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

        <element name="JurisdictionRef" type="mpeg7:ReferenceType"/>
      </choice>
      <choice minOccurs="0">
        <element name="Address" type="mpeg7:PlaceType"/>
        <element name="AddressRef" type="mpeg7:ReferenceType"/>
      </choice>
      <element name="ElectronicAddress" type="mpeg7:ElectronicAddressType"
        minOccurs="0"/>
    </sequence>
  </extension>
</complexContent>
</complexType>

```

7.5.4.3 PersonGroup DS semantics

Semantics of the PersonGroupType:

Name	Definition
PersonGroupType	Describes a group of persons.
Name	Describes a name by which the group is known. A group may have multiple names.
type	Indicates the type of name using one of the following values: <i>former</i> – The name is no longer used. <i>variant</i> – The name is a variation of the official or most commonly used name. For example, an abbreviated form of the official name, or an informal nickname. <i>main</i> – The name is either the official one or is widely known and used.
NameTerm	Describes a name by which the group is known using a term from a classification scheme.
type	Indicates the type of name using one of the following values: <i>former</i> – The name is no longer used. <i>variant</i> – The name is a variation of the official or most commonly used name. For example, an abbreviated form of the official name, or an informal nickname. <i>main</i> – The name is either the official one or is widely known and used.
Kind	Indicates the kind of group (e.g. "rock band," "ballet ensemble," and so forth), which may be expressed using a controlled term.
Member	Describes a person who is a member of the group.
MemberRef	Describes a reference to a person who is a member of the group.
Jurisdiction	Describes the place that corresponds to the jurisdiction under which this group is entered.
JurisdictionRef	Describes a reference to the place that corresponds to the jurisdiction under which this group is entered.
Address	Describes a place that is the address where this group is located.

Name	Definition
AddressRef	Describes a reference to a place that is the address where this group is located.
ElectronicAddress	Describe the electronic address information for this group.

7.5.4.4 PersonGroup DS examples (informative)

The following example describes the person group artist of musical content. The example shows how the PersonGroup describes a musical group, the Rolling Stones. Each member of the group (Mick Jagger, Keith Richard, Ron Wood, and Charlie Watts) is represented as a Member of the group. Each Member is represented by an instance of the Person DS.

```
<Mpeg7>
  <Description xsi:type="CreationDescriptionType">
    <CreationInformation>
      <Creation>
        <Title>Beggars Banquet</Title>
        <Creator>
          <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:ARTIST"/>
          <Agent xsi:type="PersonGroupType">
            <Name>The Rolling Stones</Name>
            <Kind>
              <Name>Rock band</Name>
            </Kind>
            <Member>
              <Name>
                <GivenName initial="M">Mick</GivenName>
                <FamilyName>Jagger</FamilyName>
              </Name>
            </Member>
            <Member>
              <Name>
                <GivenName>Keith</GivenName>
                <FamilyName>Richard</FamilyName>
              </Name>
            </Member>
            <Member>
              <Name>
                <GivenName>Ron</GivenName>
                <FamilyName>Wood</FamilyName>
              </Name>
            </Member>
            <Member>
              <Name>
                <GivenName>Charly</GivenName>
                <FamilyName>Watts</FamilyName>
              </Name>
            </Member>
          </Agent>
        </Creator>
      </Creation>
    </CreationInformation>
  </Description>
</Mpeg7>
```

7.5.5 Organization DS

7.5.5.1 Introduction

The Organization DS describes an organization: a group of persons with a collective title whose member persons are not easily identified or whose membership is very large, i.e. professional associations, corporate entities, production companies, unions, government bodies, and so forth.

7.5.5.2 Organization DS syntax

```

<!-- ##### -->
<!-- Definition of Organization DS (7.5.5) -->
<!-- ##### -->

<!-- Definition of Organization DS -->
<complexType name="OrganizationType">
  <complexContent>
    <extension base="mpeg7:AgentType">
      <sequence>
        <element name="Name" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="mpeg7:TextualType">
                <attribute name="type" use="optional">
                  <simpleType>
                    <restriction base="NMTOKEN">
                      <enumeration value="former"/>
                      <enumeration value="variant"/>
                      <enumeration value="main"/>
                    </restriction>
                  </simpleType>
                </attribute>
              </extension>
            </simpleContent>
          </complexType>
        </element>
        <element name="NameTerm" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:ControlledTermUseType">
                <attribute name="type" use="optional">
                  <simpleType>
                    <restriction base="NMTOKEN">
                      <enumeration value="former"/>
                      <enumeration value="variant"/>
                      <enumeration value="main"/>
                    </restriction>
                  </simpleType>
                </attribute>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="Kind" type="mpeg7:TermUseType" minOccurs="0"/>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Contact" type="mpeg7:AgentType"/>
          <element name="ContactRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0">

```

```

        <element name="Jurisdiction" type="mpeg7:PlaceType"/>
        <element name="JurisdictionRef" type="mpeg7:ReferenceType"/>
    </choice>
    <choice minOccurs="0">
        <element name="Address" type="mpeg7:PlaceType"/>
        <element name="AddressRef" type="mpeg7:ReferenceType"/>
    </choice>
    <element name="ElectronicAddress" type="mpeg7:ElectronicAddressType"
        minOccurs="0"/>
    </sequence>
</extension>
</complexContent>
</complexType>

```

7.5.5.3 Organization DS semantics

Semantics of the OrganizationType:

Name	Definition
OrganizationType	Describes an organization.
Name	Describes a name by which the organization is known. A group may have multiple names.
type	Indicates the type of name using one of the following values: • <i>former</i> – The name is no longer used. • <i>variant</i> – The name is a variation of the official or most commonly used name. For example, an abbreviated form of the official name, or an informal nickname. • <i>main</i> – The name is either the official one or is widely known and used.
NameTerm	Describes a name by which the organization is known using a term from a classification scheme.
type	Indicates the type of name using one of the following values: • <i>former</i> – The name is no longer used. • <i>variant</i> – The name is a variation of the official or most commonly used name. For example, an abbreviated form of the official name, or an informal nickname. • <i>main</i> – The name is either the official one or is widely known and used.
Kind	Indicates the kind of organization (e.g. "company," "museum," "NGO," and so forth), which may be expressed using a controlled term.
Contact	Describes an agent who acts as a contact for the organization.
ContactRef	Describes a reference to an agent who acts as a contact for the organization.
Jurisdiction	Describes the place that corresponds to the jurisdiction under which this organization is entered.

Name	Definition
JurisdictionRef	Describes a reference to the place that corresponds to the jurisdiction under which this organization is entered.
Address	Describes a place that is the address where this organization is located.
AddressRef	Describes a reference to a place that is the address where this organization is located.
ElectronicAddress	Describes the electronic address information for this organization.

7.5.5.4 Organization DS examples (informative)

The following example shows an example description of a disseminator organization named "Acme Inc." In this example, the kind of organization is a company. The contact for the company is a person named "John Smith" who can be reached via his e-mail address of "jsmith@acme.org". Furthermore, the jurisdiction is the United States.

```

<Mpeg7>
  <Description xsi:type="CreationDescriptionType">
    <CreationInformation>
      <Creation>
        <Title/>
        <Creator>
          <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:DISSEMINATOR"/>
          <Agent xsi:type="OrganizationType" id="ORG1">
            <Name type="main" xml:lang="en">Acme Inc.</Name>
            <Name type="variant" xml:lang="en">ACME</Name>
            <Kind>
              <Name>company</Name>
            </Kind>
            <Contact xsi:type="PersonType">
              <Name>
                <GivenName>John</GivenName>
                <FamilyName>Smith</FamilyName>
              </Name>
              <ElectronicAddress>
                <Email>jsmith@acme.org</Email>
              </ElectronicAddress>
            </Contact>
            <Jurisdiction>
              <Region>us</Region>
            </Jurisdiction>
            <ElectronicAddress>
              <Url>www.acme.com</Url>
            </ElectronicAddress>
          </Agent>
        </Creator>
      </Creation>
    </CreationInformation>
  </Description>
</Mpeg7>

```

7.5.6 PersonName datatype

7.5.6.1 Introduction

The PersonName datatype describes the name of a person, whether historical, existing, or fictional.

7.5.6.2 PersonName datatype syntax

```
<!-- ##### -->
<!-- Definition of PersonName datatype (7.5.6) -->
<!-- ##### -->

<!-- Definition of PersonName datatype -->
<complexType name="PersonNameType">
  <sequence>
    <choice minOccurs="1" maxOccurs="unbounded">
      <element name="GivenName" type="mpeg7:NameComponentType"/>
      <element name="FamilyName" type="mpeg7:NameComponentType" minOccurs="0"/>
      <element name="Title" type="mpeg7:NameComponentType" minOccurs="0"/>
      <element name="Numeration" type="string" minOccurs="0"/>
    </choice>
  </sequence>
  <attribute name="dateFrom" type="mpeg7:timePointType" use="optional"/>
  <attribute name="dateTo" type="mpeg7:timePointType" use="optional"/>
  <attribute name="type" use="optional">
    <simpleType>
      <restriction base="NMTOKEN">
        <enumeration value="former"/>
        <enumeration value="variant"/>
        <enumeration value="main"/>
      </restriction>
    </simpleType>
  </attribute>
  <attribute ref="xml:lang" use="optional"/>
</complexType>

<!-- Definition of NameComponent datatype -->
<complexType name="NameComponentType">
  <simpleContent>
    <extension base="mpeg7:TextualBaseType">
      <attribute name="initial" type="string" use="optional"/>
      <attribute name="abbrev" type="string" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

7.5.6.3 PersonName datatype semantics

Semantics of the PersonNameType:

Name	Definition
PersonNameType	Describes the name of a person.
GivenName	Describes the given name(s) of a person. At least one given name shall be specified in a name description. Given names include middle names and/or middle initials, if known. Given names may be initials or abbreviation if only these elements are known.

Name	Definition
FamilyName	Describes the family name(s) of an individual. This may be a surname, clan name, and so forth. The use of this element is optional since not all person names include a family name.
Title	Indicates the titles that are part of the name of an individual, such as honorifics. This part of a name is optional.
Numeration	Indicates a roman numeral or other numeric designation that is part of the name of an individual. For example, the "III" in "William James III."
dateFrom	Indicates the date starting from which this name came into use (e.g. date of birth, date of change to a new name, and so forth). If not specified, then no such date is known.
dateTo	Indicates the date starting from which this name ceased to be used (death, old name after name change, etc). If not specified, then use of the name is still current.
type	Indicates the type of name using one of the following values: • <i>former</i> – The name is no longer used. • <i>variant</i> – The name is a variation of the official or most commonly used name. For example, an abbreviated form of the official name, or an informal nickname. • <i>main</i> – The name is either the official one or is widely known and used.
xml:lang	Describes the language in which the name is described. This need not be the same as the original language of the name. For example, a Japanese name can be described in English.

Semantics of the NameComponentType:

Name	Definition
NameComponentType	Describes a component of a name.
initial	Indicates the initial for the name component. Initials are a truncated form of the name component.
abbrev	Indicates an alternative shorter form for a name component. For example, "Bill" for "William."

While the conventions for naming people vary widely across cultures and languages, the Person Name datatype provides a simple model for the structure of names. However it does not require that this structure be used.

The model for names is as follows. First, it is assumed that everyone is named by some given name. Secondly, a name may contain a family name, which is shared by the members of a relevant social group, identified according to the social conventions of that culture. In many cultures this is a family name, but in some cultures it may be a place name used to identify the place where a person comes from or lives in. In some cultures, no such name exists and there is only a given name. Finally, a title indicates a part of a name conferred upon a person as result of obtaining a certain social status, such as "Sir" for knighthood or "Ph.D." for a degree.

The elements of a name may also be broken down into name components. However, this is entirely optional. The decomposition of a name is culture and language-bound; however, each component usually corresponds to one word in the language of the name. Furthermore, abbreviations and an initial can be described. The abbreviation refers to a variant form of the name such as a nickname or pet name like "Johnny" for "Jonathon." The initial is a shorter string used to denote a name component. For example, an initial can be the first letter of a name in English. Abbreviated names or initials may be used if the full form is unknown.

The order in which components of a name appear in the description is significant and should adhere to the order used for writing names in the language in which they are described. For example, a Japanese name should appear with the `FamilyName` first and then the `GivenName(s)`. On the other hand, an English name should appear with `GivenName(s)` first, followed by the `FamilyName`.

A name is not assumed to be unique identifier for a person, and the description is not in a canonical form, so there may be several different ways to describe the same name using the `PersonName` datatype. More than one `PersonName` can be associated with a person when that person is known by more than one name.

7.5.6.4 `PersonName` datatype examples (informative)

The following examples of description excerpts show the use of `PersonName` datatype for describing the name for "Edward Hoover." In the first example, the name is represented in the simplest fashion without any decomposition:

```
<PersonName>
  <GivenName>Edward J. Hoover</GivenName>
</PersonName>
```

The following example describes the name in a more structured fashion:

```
<PersonName>
  <GivenName>Edward</GivenName>
  <GivenName initial="J"/>
  <FamilyName>Hoover</FamilyName>
</PersonName>
```

The following example describes the name in more details by decomposing the name into components:

```
<PersonName>
  <Title>Professor</Title>
  <GivenName>Edward</GivenName>
  <GivenName initial="J">Jermina</GivenName>
  <FamilyName>Hoover</FamilyName>
  <FamilyName>van Blumen</FamilyName>
  <FamilyName initial="Jr.">Junior</FamilyName>
  <Title>F.R.S.</Title>
</PersonName>
```

The following example shows a name for which only a given name exists:

```
<Name>
  <GivenName>Dinesh</GivenName>
</Name>
```

The following examples describes a Spanish name in which the given name cannot be broken down further.

Note the use of the language attribute `xml:lang = "es"` to describe the language of the name as Spanish.

```

<Name xml:lang="es">
  <GivenName initial="J.M">Jose Maria</GivenName>
  <FamilyName>Garcia</FamilyName>
  <FamilyName>Sanchez</FamilyName>
</Name>

```

7.5.6.5 PersonName datatype usage guidelines (informative)

The following guidelines should be followed for describing names:

- **Common name:** A name should be selected by which the individual is most commonly known. Sources of information in the language of the individual should be used to determine a real name, pseudonym, title of nobility, nickname, initials, or another name. If more than one name is used, the most commonly known name should be used.
- **Library practice:** Information sources such as the Library of Congress Name Authority Files should be consulted prior to establishing the name in local files. In the absence of access to national data, local decisions about the predominant form of names based on resources available locally, and associated name references for variant names shall be "linked" to the official name. Variations in spelling, different transliteration systems from non-roman alphabets, names in different languages, fuller versus abbreviated forms of name, or changes in the name used provide opportunities for linking to the official form of name.
- **Separate entities:** When a person actively uses two or more names for different types of creative works, separate entries can be created. Example: Dodgson, Charles Ludwidge (for mathematical works) and Carroll, Lewis (literary works). Entries would be created under both names with reference to the other name.
- **Change of name:** When names change due to marriage, the latest form of name should be chosen unless it is determined that an earlier name persists as the name by which a person is better known.
- **Qualification elements:** Qualification elements such as dates, titles of address, fuller form of name, and so forth, should be added to distinguish between persons with the same name.
- **Multilingual:** For persons using more than one language the form corresponding to the language of most of the works should be chosen. The other language forms should be linked by referencing. The name most commonly found in reference sources of the person's country of residence or activity should be used. The country of birth need not be the country of the person's residence or predominant creative activity. Names utilized in professional contribution should be used, with references from the additional names.

7.5.7 Electronic Address datatype

7.5.7.1 Introduction

The ElectronicAddress datatype describes electronic address information, including telephone number, fax number, e-mail address, and URLs.

7.5.7.2 ElectronicAddress datatype syntax

```

<!-- ##### -->
<!-- Definition of ElectronicAddress datatype (7.5.7) -->
<!-- ##### -->

<!-- Definition of ElectronicAddress datatype -->
<complexType name="ElectronicAddressType">
  <sequence>
    <element name="Telephone" type="string" minOccurs="0" maxOccurs="unbounded"/>
    <element name="Fax" type="string" minOccurs="0" maxOccurs="unbounded"/>
    <element name="Email" type="string" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

```



```

    <element name="Url" type="anyURI" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

```

7.5.7.3 ElectronicAddress datatype semantics

Semantics of the ElectronicAddressType:

Name	Definition
ElectronicAddressType	Describes an electronic address.
Telephone	Describes a telephone number.
Fax	Describes a fax number.
Email	Describes an E-mail address.
Url	Describes a URL address, such as a home page.

7.6 Description of places

7.6.1 Introduction

This subclause specifies tools for describing places appearing in multimedia content or relating to multimedia content.

7.6.2 Place DS

7.6.2.1 Introduction

The Place DS describes existing, historical, and fictional places. The Place DS can be used to describe places depicted in the content, such as the setting, and places in the real world, such as where the content was filmed.

7.6.2.2 Place DS syntax

```

<!-- ##### -->
<!-- Definition of Place DS (7.6.2) -->
<!-- ##### -->

<!-- Definition of Place DS -->
<complexType name="PlaceType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Name" type="mpeg7:TextualType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="NameTerm" type="mpeg7:ControlledTermUseType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="Role" type="mpeg7:TermUseType"
          minOccurs="0"/>
        <element name="GeographicPosition" minOccurs="0">
          <complexType>
            <sequence>

```

```

        <element name="Point" type="mpeg7:GeographicPointType"/>
    </sequence>
    <attribute name="datum" type="string" use="optional"/>
</complexType>
</element>
<element name="AstronomicalBody" type="mpeg7:TermUseType"
    minOccurs="0" maxOccurs="unbounded"/>
<element name="Region" type="mpeg7:regionCode"
    minOccurs="0" maxOccurs="unbounded"/>
<element name="AdministrativeUnit"
    minOccurs="0" maxOccurs="unbounded">
    <complexType>
        <simpleContent>
            <extension base="string">
                <attribute name="type" type="string" use="optional"/>
            </extension>
        </simpleContent>
    </complexType>
</element>
<element name="PostalAddress" minOccurs="0">
    <complexType>
        <sequence>
            <element name="AddressLine" type="mpeg7:TextualType"
                minOccurs="1" maxOccurs="unbounded"/>
            <element name="PostingIdentifier" type="mpeg7:TextualType"
                minOccurs="0"/>
        </sequence>
        <attribute ref="xml:lang" use="optional"/>
    </complexType>
</element>
<element name="InternalCoordinates" type="string" minOccurs="0"/>
</sequence>
<attribute ref="xml:lang" use="optional"/>
</extension>
</complexContent>
</complexType>

```

7.6.2.3 Place DS semantics

Semantics of the PlaceType:

Name	Definition
PlaceType	Describes a location.
Name	Describes a name of the place. Multiple values are allowed to support multiple languages.
NameTerm	Describes a name of the place using a term from a classification scheme.
Role	Indicates the role of the place, such as "real location," "fictional setting," or "shooting location."
GeographicPosition	Describes the geographic position of the place.
Point	Describes a geographic point in the place.

Name	Definition
datum	Indicates the geodetic datum system used in the description of GeographicPosition (optional). Possible values include "wgs84" (World Geodetic System 1984), "itrf" (International Terrestrial Reference Frames), "tokyo" (Tokyo Datum) and so on.
AstronomicalBody	Describes the name of the astronomical body on which the location exists. For example, "Earth."
Region	Describes the region of the place specified using the appropriate ISO 3166-2 region code.
AdministrativeUnit	Describes the name of the administrative units to which the place belongs-e.g. a city, a town, a village, or even a fictional unit such as the USS Enterprise. The administrative unit is a place within the region specified. More than one administrative unit may be specified. In this case the order of administrative is interpreted as being from largest to smallest. In other words, each administrative unit should contain the subsequent administrative unit.
type	Indicates the type of administrative unit being identified (e.g. "town," "city," "spaceship").
PostalAddress	Describes the postal address including street (or square) and house number or name of the place. The postal address does not need to include the information specified by other elements of the Place DS, such as country, region, or administrative unit. However, such information may be included in this element.
AddressLine	Describes one line of the postal address of the place. The division of the postal address into lines is not specified by this standard.
PostingIdentifier	Describes the postal identifier for the place (optional). For example, a ZIP code or Postal code.
InternalCoordinates	Describes the internal coordinates of the place within the specified postal address (e.g. an apartment or room number, the drawing room, etc.).
xml:lang	Indicates the language used to describe the place.

7.6.2.4 Place DS examples (informative)

The following example shows the use of Place DS for describing a shooting location. In this case, the place is a university in Madrid, Spain.

```
<Mpeg7>
  <DescriptionUnit xsi:type="PlaceType">
    <Name xml:lang="en">Madrid</Name>
    <Role>
      <Name>shooting location</Name>
    </Role>
    <GeographicPosition datum="itrf">
      <Point longitude="135.75" latitude="35.5" altitude="100"/>
    </GeographicPosition>
    <Region>es-cam</Region>
    <AdministrativeUnit type="city">Madrid</AdministrativeUnit>
    <PostalAddress>
      <AddressLine>E.T.S.Ing. Telecommunication</AddressLine>
      <AddressLine>Universidad Politecnica de Madrid</AddressLine>
    </PostalAddress>
  </DescriptionUnit>
</Mpeg7>
```

```

        <AddressLine>Ciudad Universitaria s/n</AddressLine>
        <PostingIdentifier>E-28040</PostingIdentifier>
    </PostalAddress>
    <InternalCoordinates>C-306</InternalCoordinates>
</DescriptionUnit>
</Mpeg7>

```

7.6.3 GeographicPoint datatype

7.6.3.1 Introduction

The GeographicPointType datatype describes a geographic point using coordinates in some Global Positioning System.

7.6.3.2 GeographicPoint datatype syntax

```

<!-- ##### -->
<!-- Definition of GeographicPoint datatype (7.6.3) -->
<!-- ##### -->

<!-- Definition of GeographicPoint datatype -->
<complexType name="GeographicPointType">
  <attribute name="longitude" use="required">
    <simpleType>
      <restriction base="double">
        <minInclusive value="-180.0"/>
        <maxInclusive value="180.0"/>
      </restriction>
    </simpleType>
  </attribute>
  <attribute name="latitude" use="required">
    <simpleType>
      <restriction base="double">
        <minInclusive value="-90.0"/>
        <maxInclusive value="90.0"/>
      </restriction>
    </simpleType>
  </attribute>
  <attribute name="altitude" type="double" use="optional"/>
</complexType>

```

7.6.3.3 GeographicPoint datatype semantics

Semantics of the GeographicPointType:

Name	Definition
GeographicPointType	Describes a geographic point provided via its coordinates in some Global Positioning System.
longitude	Describes the longitude in degrees. Negative values represent western longitudes.
latitude	Describes the latitude in degrees. Negative value represent southern latitude.
altitude	Describes the altitude in meters. The reference altitude, indicated by zero, of the measurement is set to the sea level as default. If the geodetic datum system used to measure the altitude is specified elsewhere, the datum point is determined by

Name	Definition
	the system specification. Negative values indicate altitudes below the reference altitude.

7.6.3.4 GeographicPoint datatype examples (informative)

The following example description excerpt shows the use of GeographicPoint datatype for describing a geographical point which is at longitude = 135.75 and latitude = 35.5 and altitude = 100.

```
<GeographicPosition>
  <Point longitude="135.75" latitude="35.5" altitude="100"/>
</GeographicPosition>
```

7.7 Graphs and relations

7.7.1 Introduction

This subclause specifies tools for describing relations between description tool instances and for structuring descriptions using graphs.

7.7.2 Graph DS

7.7.2.1 Introduction

The Graph DS describes a graph of relations amongst a set of description scheme instances; for example, a graph describing the narrative structure of a movie or the spatial structure of a set of segments.

7.7.2.2 Graph DS syntax

```
<!-- ##### -->
<!-- Definition the Graph DS (7.7.2) -->
<!-- ##### -->

<!-- Definition of Graph DS -->
<complexType name="GraphType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="Node">
          <complexType>
            <complexContent>
              <extension base="mpeg7:ReferenceType">
                <sequence>
                  <element name="Name" type="mpeg7:TermUseType"
                    minOccurs="0"/>
                </sequence>
                <attribute name="id" type="ID" use="optional"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="Relation" type="mpeg7:RelationType"/>
      </choice>
```

```

</extension>
</complexContent>
</complexType>

```

7.7.2.3 Graph DS semantics

Semantics of the GraphType.

Name	Definition
GraphType	Describes a set of description organized into a graph structure.
Node	Describes a graph node. The node references the DS corresponding to this node
Name	Indicates the type of entity represented by this node using a term. This mechanism is used in graph classification schemes, see the examples in clause 7.4.15.
id	Identifies the id of the node (optional). The id attribute shall be used in references and relations that refer to this node.
Relation	Describes a relation between one or more nodes.

Nodes are optional. Any item referenced by a relation in a graph is automatically defined as a node (this is called an anonymous node). For example, the following two representations are equivalent up to isomorphism.

```

<Graph>
  <Node id="n1" idref="a">
    <Name href="urn:example:bicycle"/>
  </Node>
  <Node id="n2" idref="b">
    <Name href="urn:example:object"/>
  </Node>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:specializes"
    source="#n1" target="#n2"/>
</Graph>

<Graph>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:specializes"
    source="#a" target="#b"/>
</Graph>

```

7.7.2.4 Graph DS formal semantics (informative)

The mathematical object represented by the Graph DS is called a polyadic graph. The rest of this subclause formalizes the properties of the Graph DS using this notion. The subclause also contains useful formal definitions of related graphs, and of labels, which are used implicitly in the definition of graphs using Graph DS.

Definition: A graph $G = \{E, V\}$ is an ordered pair of two sets, V a set of vertices, and $E \subseteq V \times V$ a set of relations (also called edges).

Note: in this definition a relation is an ordered pair; therefore, each has an associated "direction". Such graphs are also called directed graphs or simply digraphs.

Definition: If f is a relation, the set of all ordered pairs $(x, y) \in f$ is called the graph of f .

Definition: A generalization of a graph where the edges involved are n -ary relations is called a hypergraph. Formally, a hypergraph $G = \langle E, V \rangle$ is an ordered pair of two sets, V , a set of vertices, and $E \subseteq V \times \dots \times V$ a set of edges.

Definition: A bipartite graph is a graph in which the set of vertices V can be divided into two parts, X , Y such that all relations are of the form $e(x,y)$ or $e(y,x)$.

Definition: Given a set A , one may treat A as "letters" in an alphabet, and construct "words" from it by ordered lists of elements of A . Call such a "word" a string in A , in keeping with the terminology in computer science literature. The length of the string is the number of "letters," if it were looked at as a set, this would be the cardinality of the string, if looked at as an ordered n -tuple, it is the length of the n -tuple.

Definition: For any set A , define the set of strings constructed from elements of A as the set A^* .

Definition: For any set A define the set of non-empty strings constructed from elements of A as the set $A^+ \subseteq A^*$.

Definition: A polyadic, or n -ary graph is an ordered pair of two sets (E, V) together with two maps s , and t (representing the source and target), $\langle s, t \rangle: E \rightarrow V^* \times V^*$. Thus if $s(e)$ is a string of length m and $t(e)$ is a string of length n , then e is an relation between groups of m elements and groups of n elements. Formally, this is called an element in an (m,n) -ary relation. A binary relation is thus composed of $(1,1)$ -ary elements, so relations in this document are expressed as $m+n$ -ary.

Note: the definition also implies the existence of unary relations, which are formally $(0,1)$ -ary. In the relations in this document, an (m,n) -ary relation has m source elements, and n target elements, and a unary relation has only a target.

Definition: A graph in which to each vertex v there is a relation $e(v,v)$, is called reflexive. A graph that has at least one relation connecting any two vertices is called complete. A reflexive graph which has no relations not of the form $e(v,v)$, is called discrete.

Definition: A labeled (hyper)graph is a graph G is 4-tuple $G = \langle E, V, L, l \rangle$ where E and V are as before and L is a set of labels (an alphabet) and $l: E \rightarrow L$ is a function assigning a label to each relation.

There are many ways to label a graph. One is to split each edge into two edges and insert a node corresponding to the label. Another is to stipulate a function mapping each edge in the edge set to a set of points in another set called the set of labels, or the alphabet. Yet another is to specify a graph for an alphabet, and map all nodes and edges into those in the alphabet graph. Since this labels both the edges and the vertices, and specifies which combinations may be used with which labels, this is the most general, and the clearest representation. For instance, for UML, one defines nodes like parent, child, container, contained, and so forth, and the edge between child and parent is called inheritance, that between contained and container is called aggregation, and so forth

Definition: A graph morphism between graphs G and H is a pair F of mappings, $f: V(G) \rightarrow V(H)$ and $g: E(G) \rightarrow E(H)$ with the property that $g(e(u,v)) = e(f(u), f(v))$, i.e. that the image of an relation between two vertices is an relation between the images of the vertices.

A labeling in its simplest form can then be a graph morphism between a graph and a discrete graph, which the nodes in the latter define the labels. More complex labeling by graphs are also called structures, colors, and types. Relationships can be defined in the way in which they are most frequently described in social or object terms:

Definition: A path P between two vertices u, v in a directed graph is a list of nodes and edges which connect them together in the following sense: 1) there is a relation $e(u, t)$ and a relation $e(s, v)$ for some $s, t \in V$, 2) for each $e(s, t) \in P$ either $s=u$, $v=t$, or there are relations $e(r, s), e(t, w)$ also in P .

Definition: Two vertices u, v in a graph G are related, if there exists a path $P(u, v)$ between them. A relationship between u and v is a collection of paths between them.

7.7.2.5 Graph DS examples (informative)

The following example illustrates the use of Graph DS for describing the graph of relations between two objects. The following relations are described: left and goalOf, which means that object A is to the left of object B , and object A is the goal of object B . Additional examples of the use of the Graph DS are given in 11.10.4 and 12.5.2.4.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Graph of relations between object A and object B </Name>
      </Label>
      <SemanticBase xsi:type="ObjectType" id="objectA">
        <Label>
          <Name> Object A </Name>
        </Label>
      </SemanticBase>
      <SemanticBase xsi:type="ObjectType" id="objectB">
        <Label>
          <Name> Object B </Name>
        </Label>
      </SemanticBase>
      <Graph>
        <Node id="nodeA" href="#objectA"/>
        <Node id="nodeB" href="#objectB"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:left"
          source="#nodeA" target="#nodeB"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:goalOf"
          source="#nodeA" target="#nodeB"/>
      </Graph>
    </Semantics>
  </Description>
</Mpeg7>

```

7.7.3 Relation DS

7.7.3.1 Introduction

The Relation DS describes a relation amongst a set of description scheme instances. The Relation DS supports two different kinds of use:

1. **Internal Relations.** The relation is written *inside* the description for the first source argument. The first source argument is implicitly identified as the XML element containing the instance of the Relation DS. The source attribute shall not be specified in this case.
2. **External Relations.** The relation is written *outside* of the description scheme instances being related. All arguments are specified explicitly by giving their references.

7.7.3.2 Relation DS syntax

```

<!-- ##### -->
<!-- Definition the Relation DS (7.7.3) -->
<!-- ##### -->

<!-- Definition of Relation DS -->
<complexType name="RelationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <attribute name="type" type="mpeg7:termReferenceType" use="optional"/>
      <attribute name="source" use="optional">
        <simpleType>
          <list itemType="anyURI"/>
        </simpleType>
      </attribute>
    </extension>
  </complexContent>
</complexType>

```



```

    <attribute name="target">
      <simpleType>
        <list itemType="anyURI"/>
      </simpleType>
    </attribute>
    <attribute name="directed" type="boolean"
      use="optional" default="true"/>
    <attribute name="strength" type="mpeg7:zeroToOneType"
      use="optional" default="1.0"/>
  </extension>
</complexContent>
</complexType>

```

7.7.3.3 Relation DS semantics

Semantics of the RelationType:

Name	Definition
RelationType	Describes a relation among a set of DS instances. The arguments of a relation are categorized into source (input) and target (output) arguments. Both the source and target arguments for a relation are ordered. A relation may have a variable number of arguments.
type	Identifies the relation uniquely using a reference to a term in a classification scheme defining this relation.
source	Describes references to the DS instances that are the source argument(s) of the relation. The URI references contained in this attribute are ordered so that the first URI identifies the first source argument, the second item the second argument and so on.
target	Describes references to the DS instances that are the target argument of the relation. The URI references contained in this attribute are ordered so that the first URI identifies the first target argument, the second item the second argument, and so on.
strength	Indicates the strength of the relationship on a scale from [0,1], where one is the strongest value and zero the weakest (optional). This can be used to represent fuzzy graphs. The default value is one.
directed	Indicates that the relation is directed (optional). That is, the arguments of the relation are distinguished as source and target arguments. For undirected relations no such distinction is made. The default value is true.

7.7.3.4 Formal Semantics

Definition: An n -ary relation R on n collections $\{A_1, \dots, A_n\}$ is a subset of their Cartesian product, i.e. $R \subseteq \times A_i$.

Definition: A binary relation R on two collections A, B , is a subset of the Cartesian product $R \subseteq A \times B$. Denote $(a, b) \in R$ by aRb .

Definition: Several specialized names are used for binary relations, to describe properties of the collection R :

- *Reflexive.* A binary relation such that for all a , aRa is called reflexive.
- *Symmetric.* If $aRb \Rightarrow bRa$, the relation is called symmetric.
- *Anti-symmetric.* If aRb and bRa implies $a=b$ then the relation is called anti-symmetric.

- *Transitive*. If aRb and bRc imply that aRc the relation is called transitive.
- *Function*. Given a relation between two sets, $f \subseteq X \times Y$, such that for each x there is a unique y with xfy , then f is called a function, and normally $f(x)=y$.
- *Total*. If for every pair $a, b \in A$, either aRb or bRa then the relation is total.

Definition: The following special binary relation types are defined and used in this standard:

- *Pre-order*. A relation that is transitive and reflexive is called a pre-order.
- *Partial Order*. If a relation is a pre-order and it is also anti-symmetric, then it is called a partial order.
- *Total (Linear) Order*. Given a partial order $R \subseteq A \times A$ for which, given $a, b \in A$, either aRb or bRa holds, it is called a total order. A linear order is a total order that is not reflexive.
- *Equivalence Relation*. A relation that is symmetric, reflexive, and transitive is called an equivalence relation.

7.7.3.5 Relation DS examples (informative)

The following example shows the use of the Relation DS for describing relations. Both of the following examples are equivalent in describing the relations that segment A is to the left of segment B and differ only in their syntax. The graph with two nodes and one labeled edge leftOf describes the relation as follows:

```
<Graph>
  <Node id="nodeA1" idref="segmentA"/>
  <Node id="nodeB1" idref="segmentB"/>
  <!-- Edge from a->b -->
  <Relation type="urn:relation:leftOf" source="#nodeA1" target="#nodeB1"/>
</Graph>
```

Alternatively, the Relation DS describes the relation within the source argument (VideoSegment "segmentA") as follows:

```
<VideoSegment id="segmentA">
  <Relation type="urn:relation:leftOf" target="#segmentB"/>
</VideoSegment>
<VideoSegment id="segmentB"/>
```

7.7.4 MorphismGraph DS

7.7.4.1 Introduction

The MorphismGraph DS represents a mapping between a source and a target graph, such that the mapping preserves the edge and node structure (i.e. if an edge in the source graph is mapped to an edge h in the target graph, then the source and target nodes of e are mapped to the source and target nodes of h). The mapping establishes the correspondence between the nodes and relations of a source graph and the nodes and relations of a target graph. The correspondences are represented as relations in the morphism graph.

7.7.4.2 MorphismGraph DS syntax

```
<!-- ##### -->
<!-- Definition the MorphismGraph DS (7.7.4) -->
<!-- ##### -->
```

```

<!-- Definition of MorphismGraph DS -->
<complexType name="MorphismGraphType">
  <complexContent>
    <extension base="mpeg7:GraphType">
      <sequence>
        <choice>
          <element name="SourceGraphRef" type="mpeg7:ReferenceType"/>
          <element name="SourceGraphTermRef"
            type="mpeg7:ControlledTermUseType"/>
        </choice>
        <choice>
          <element name="TargetGraphRef" type="mpeg7:ReferenceType"/>
          <element name="TargetGraphTermRef"
            type="mpeg7:ControlledTermUseType"/>
        </choice>
        <element name="RuleTermRef"
          type="mpeg7:ControlledTermUseType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

7.7.4.3 MorphismGraph DS semantics

Semantics of the MorphismGraphType:

Name	Definition
MorphismGraphType	Describes a morphism between a source and a target graph. MorphismGraphType extends GraphType.
SourceGraphRef	Describes a reference to the source graph of the morphism.
SourceGraphTermRef	Describes a reference to a graphical term, in a graphical classification scheme, that defines the source graph of the morphism. For production rules, this element points to the LHSGraph in both the PushoutDefinition and DoublePushoutDefinition. Pointing to another graph in these rules requires the use of SourceRef.
TargetGraphRef	Describes a reference to the target graph of the morphism.
TargetGraphTermRef	Describes a reference to the graphical term, in a graphical classification scheme, that defines the target graph of the morphism. TargetTermRef. For production rules, this element points to the AlphabetGraph in the PullbackDefinition and to the first RuleGraph in the DoublePullbackDefinition. Pointing to another graph in these rules requires the use of TargetRef.
RuleTermRef	Describes a reference to a term defining a graphical production being applied by this morphism graph (optional).

7.7.4.4 MorphismGraph DS examples (informative)

The following example description excerpt shows the use of the MorphismGraph DS for describing a morphism graph.

```
<MorphismGraph>
  <Relation id="left1" type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"
    source="#object1" target="#object3"/>
  <Relation id="left2" type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"
    source="#object2" target="#object4"/>
  <SourceGraphRef href="#context"/>
  <TargetGraphRef href="#LHS"/>
</MorphismGraph>
```

7.7.5 Graph relations classification scheme

7.7.5.1 Introduction

This subclause defines normative relations between elements of a graph. These relations are used to establish the mappings between edges and nodes of the source and target graphs in a graph morphism.

7.7.5.2 GraphRelation CS syntax

```
<!-- ##### -->
<!-- Definition of GraphRelation CS (7.7.5) -->
<!-- ##### -->

<ClassificationScheme uri="urn:mpeg:mpeg7:cs:GraphRelationCS:2001">
  <Term termID="identity">
    <Definition xml:lang="en">
      The two items related are identical.
    </Definition>
  </Term>
  <Term termID="equivalent">
    <Definition xml:lang="en">
      The two items related are equivalent according to some partitioning
      of classes.
    </Definition>
  </Term>
  <Term termID="similar">
    <Definition xml:lang="en">
      The two items related are similar.
    </Definition>
  </Term>
  <Term termID="instance">
    <Definition xml:lang="en">
      The source item is an instance of a class.
    </Definition>
  </Term>
</ClassificationScheme>
```

7.7.5.3 GraphRelation CS semantics

Semantics of the GraphRelationCS:

Name	Definition
GraphRelationCS	Describes relations between elements of a graph. These relations are used to establish the mappings between edges and nodes of the source and target graphs in a graph morphism.
identity	Describes the relation between two that are identified with one another. This relation is its own inverse.
equivalent	Describes the relation between two nodes that are considered equivalent to one another. This relation is symmetric.
similar	Describes the relation between two nodes that are considered similar to one another. This relation is symmetric but need not be transitive. I.e. <i>A similar B</i> and <i>B similar C</i> does not necessarily imply that <i>A similar C</i> .
instance	Describes the relation between two nodes where the source argument is an instance of class identified by the second argument.

7.7.5.4 GraphRelation CS examples (informative)

The following example description excerpt shows the use of the GraphRelation CS for describing graph relations within a morphism graph.

```
<Graph xsi:type="MorphismGraphType">
  <Relation source="#object3" target="#tom1"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#object4" target="#mary1"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <Relation source="#lhs1" target="#ew1"
    type="urn:mpeg:mpeg7:cs:GraphRelationCS:2001:identity"/>
  <SourceGraphTermRef href="urn:example:cs:Graph2:1"/>
  <TargetGraphRef href="#eastwestgraph"/>
</Graph>
```

7.7.6 Base relation classification scheme

7.7.6.1 Introduction

The BaseRelation CS defines set-theoretic and topological relations amongst description tool instances. It includes relations describes different topological relations (i.e. contains, touches) which can be used for both segment and semantic entities, and set-theoretic relations like union and intersection, which are application to many different kinds of description tools (e.g. collections or semantic entities). In the following specification of the BaseRelation CS, symbols *B* and *E* are used to refer to the source of the relation and symbols *C* and *O* are used to refer to the target of the relation.

7.7.6.2 BaseRelation CS syntax

```
<!-- ##### -->
<!-- Definition of BaseRelation CS (7.7.6) -->
<!-- ##### -->
```

```

<ClassificationScheme uri="urn:mpeg:mpeg7:cs:BaseRelationCS:2001">
  <Term termID="equals">
    <Definition>
      B equals C if and only if B is equal to C.
    </Definition>
  </Term>
  <Term termID="inside">
    <Definition>
      B1, B2, ..., Bn is inside C if and only if the union of B1, ..., Bn
      is a subset of C
    </Definition>
  </Term>
  <Term termID="contains">
    <Definition> Inverse relation of inside </Definition>
  </Term>
  <Term termID="covers">
    <Definition>
      B1, B2, ..., Bn covers argument C if and only if
      the union of B1, B2, ..., Bn contains C AND that union is not equal to C
    </Definition>
  </Term>
  <Term termID="coveredBy">
    <Definition> Inverse relation of covers </Definition>
  </Term>
  <Term termID="overlaps">
    <Definition>
      B overlaps C if and only if B intersect C has non-empty interior
    </Definition>
  </Term>
  <Term termID="touches">
    <Definition>
      B1, B2, ..., Bn touches argument C if and only if B union C is is
      connected
    </Definition>
  </Term>
  <Term termID="disjoint">
    <Definition>
      B is disjoint with C if and only if the intersection of B and C is empty.
    </Definition>
  </Term>
  <Term termID="separated">
    <Definition>
      E is separated from O if and only if the intersection of E and
      the closure of A is empty AND the intersection of the closure of
      E with O is empty.
    </Definition>
  </Term>
  <Term termID="union">
    <Definition>
      A0 is the union of A1, A2, ..., An if and only if
      A0 is equal to the union of the sets A1, A2, ..., An.
    </Definition>
  </Term>
  <Term termID="intersection">
    <Definition>
      A0 is the intersection of A1, A2, ..., An if and only if
      A0 is equal to the intersection of the sets A1, A2, ..., An.
    </Definition>
  </Term>

```

```

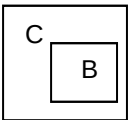
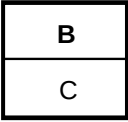
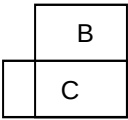
</Term>
<Term termID="member">
  <Definition>
    B is a member of C if and only if the B contains a member C.
  </Definition>
</Term>
<Term termID="memberOf">
  <Definition> Inverse relation of member </Definition>
</Term>
<Term termID="refines">
  <Definition>
    B refines C if and only if B adds detail to the description of C.
  </Definition>
</Term>
<Term termID="refinedBy">
  <Definition> Inverse relation of refines </Definition>
</Term>
</ClassificationScheme>

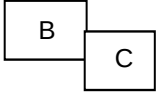
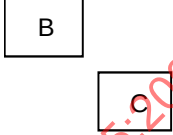
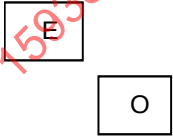
```

7.7.6.3 BaseRelation CS semantics

For each normative topological and set-theoretic relation, Table 11 includes the name, the inverse relation, the formal definition, the properties, and informative examples.

Table 11 — Formal definition, inverse relations and properties of the normative topological and set-theoretic relations with informative examples

Relation Name	Inverse Relation	Definition	Properties	Informative Examples
equals	equals	B equals C if and only if $B = C$	Equivalence	
inside	contains	$B_1, B_2, \dots B_n$ inside C if and only if $(B_1, B_2, \dots B_n) \subseteq C$	Partial order	
covers	coveredBy	$B_1, B_2, \dots B_n$ covers C if and only if $B_1 \cup B_2 \cup \dots \cup B_n \cup C \supset C$ AND $(B_1 \cup B_2 \cup \dots \cup B_n \cup C) \neq C$	Transitive	
overlaps	overlaps	B overlaps C if and only if $B \cap C$ has non-empty interior	Symmetric	

Relation Name	Inverse Relation	Definition	Properties	Informative Examples
touches	touches	$B_1, B_2, \dots B_n$ touches C if and only if $B_1 \cup B_2 \cup \dots \cup B_n \cup C$ is connected	Equivalence	
disjoint	disjoint	B disjoint C if and only if $B \cap C = \emptyset$	Symmetric	
separated	separated	E separated O if and only if $E \cap cl(O) = \emptyset$ AND $cl(E) \cap O = \emptyset$ where $cl(S)$ indicates the closure of a set S.	Symmetric	
union	-	A_0 is the union of $A_1, A_2, \dots A_n$ if and only if $A_0 = A_1 \cup A_2 \cup \dots \cup A_n$	-	-
intersection	-	A_0 is the intersection of $A_1, A_2, \dots A_n$ if and only if $A_0 = A_1 \cap A_2 \cap \dots \cap A_n$	-	-
member	memberOf	B member C if and only if $C \in B$	Pre-order	Object team member object player. Object ship memberOf object fleet.
refines	isRefinedBy	B refines C if and only if B adds detail to the meaning of C.	Pre-order	The object described as "the man with the hat" refines the object described as "the man."

7.7.6.4 BaseRelation CS examples (informative)

The following example illustrates the use of the BaseRelation CS for describing the following two relations between collections, disjoint and refines, which indicate that the collections are disjoint and that one collection refines the other collection.


```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <StructuredCollection>
        <Collection id="collection1" xsi:type="ContentCollectionType">
          <!-- more elements here -->
        </Collection>
        <Collection id="collection2" xsi:type="ContentCollectionType">
          <!-- more elements here -->
        </Collection>
        <Relationships>
          <Node id="source" href="#collection1"/>
          <Node id="target1" href="#collection2"/>
          <Relation type="urn:mpeg:mpeg7:cs:BaseRelationCS:2001:disjoint"
            source="#source" target="#target1"/>
          <Relation type="urn:mpeg:mpeg7:cs:BaseRelationCS:2001:refines"
            source="#source" target="#target1"/>
        </Relationships>
      </StructuredCollection>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

7.8 Ordering Tools

7.8.1 Introduction

This subclause defines tools for ordering descriptions according to various criteria.

7.8.2 OrderingKey DS

7.8.2.1 Introduction

The OrderingKey DS describes a hint for ordering descriptions of the same type (e.g., Segment, Object, or Event DS descriptions) using information contained within those descriptions.

7.8.2.2 OrderingKey DS syntax

```

<!-- ##### -->
<!-- Definition of OrderingKey DS (7.8.2) -->
<!-- ##### -->

<!-- Definition of OrderingKey DS -->
<complexType name="OrderingKeyType">
  <complexContent>
    <extension base="mpeg7:HeaderType">
      <sequence>
        <element name="Selector">
          <complexType>
            <attribute name="xpath" type="mpeg7:xPathSelectorType"/>
          </complexType>
        </element>
        <element name="Field" minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <attribute name="xpath" type="mpeg7:xPathFieldType"/>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

</sequence>
<attribute name="name" type="string" use="optional"/>
<attribute name="semantics" type="string" use="optional"/>
<attribute name="direction" use="optional" default="descending">
  <simpleType>
    <restriction base="NMTOKEN">
      <enumeration value="descending"/>
      <enumeration value="ascending"/>
    </restriction>
  </simpleType>
</attribute>
</extension>
</complexContent>
</complexType>

```

7.8.2.3 Ordering Key DS semantics

Semantics of the OrderingKeyType:

Name	Definition
OrderingKeyType	Describes an ordering of description tools (or parts of description tools). The OrderingKey DS is a specification of <i>hint</i> that orders instances of description tools. Note: the ordering defined in the ordering key does not constrain the actual order of appearance within the description itself.
Selector	Describes the set of description components to be ordered.
xpath	Describes an XPath expression that selects the components to be ordered in the current description. If a relative XPath expression is specified the context node of the path is the element this attribute is contained in. An absolute XPath expression is written with respect to the root of the whole description tree structure (which includes the Mpeg7 root element).
Field	Describes a reference to the parts of the selected description tools to use as a key for ordering. Field shall determine a single description part (element or attribute within the selected element) whose element content or attribute value shall be a simple type. Ordering is done on underlying value, not on its lexical representation. In addition, the simple type of the field (element or attribute value) shall be a rank-ordered datatype, e.g., a string or numeric value (integer or real) so that an alphabetical or magnitude criterion can be used in the ordering process. For example, the simple type IDREF has no associated ordering. Note: complex descriptors (or part of descriptors) shall not be used as ordering keys in order to avoid ordering ambiguities (e.g. the color histogram is not a valid ordering key because it is a vector of integers and it would not be clear for the applications what is the ordering to associate to using such a set of vector components). In case more than one Field is specified in the instantiation of this DS, the order shall adhere to the lexicographical order determined by the composite key determine by the key fields, in order of appearance in the DS instance.
xpath	Describes an XPath that identifies fields within the selected description tools to use as the key value for ordering. The XPath expression is relative to the description tools selected by the Selector XPath. A field is any of the elements and attributes of the description tool.

Name	Definition
name	Indicates the label of the OrderingKey DS description (optional).
semantics	Describes the semantics of the ordering (optional). For example, the semantics attribute of an OrderingKey DS description can be a simple word indicating the descriptor used (e.g., if it has been used the name of video segments as ordering key semantics may be "accordingtoSceneName") or a more complex textual description of the ordering results (e.g., if PanSpeed inside video segments of a soccer match is used as ordering key a possible semantics may be "Shots containing a close-up of a player")
direction	Indicates the direction of the ordering in which the DS descriptions shall be ordered. The directions of the ordering are defined as follows: <i>descending</i> - The direction of the ordering is descending order. The DS descriptions shall be ordered in descending order. <i>ascending</i> - The direction of the ordering is ascending order. The DS descriptions shall be ordered in ascending order. The attribute value is "descending" by default.

7.8.2.4 Ordering Key DS examples (informative)

The following example illustrates the use of the OrderingKey D for describing the ordering of VideoSegment DS description based on two elements of the CameraMotion D, the TRACK_LEFT_F element and the ZOOM_IN_F element. In the first OrderingKey DS description, the ordering is based on the TRACK_LEFT_F element of the CameraMotion D so the shots containing the most amount of "action" would appear first in the ordered list. In the second OrderingKey D description, the ordering is based on the ZOOM_IN_F element of the CameraMotion D so the shots containing close-ups would appear first in the ordered list.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <OrderingKey name="ActionOrdering" semantics="The shots containing more
      action">
      <Selector
        xpath="/Mpeg7/Description/MultimediaContent/TemporalDecomposition/
          VideoSegment"/>
      <Field
        xpath="VisualDescriptor/Segment/FractionalPresence/TrackLeft"/>
    </OrderingKey>
    <OrderingKey name="Close-up Ordering" semantics="The shots containing close-up
      of a player">
      <Selector
        xpath="/Mpeg7/Description/MultimediaContent/TemporalDecomposition/
          VideoSegment"/>
      <Field
        xpath="VisualDescriptor/Segment/FractionalPresence/TrackLeft"/>
    </OrderingKey>
    <MultimediaContent xsi:type="VideoType">
      <Video id="id1">
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT0M15S</MediaDuration>
        </MediaTime>
        <TemporalDecomposition>
          <VideoSegment id="id2">
            <MediaTime>

```

```

        <MediaTimePoint>T00:00:00</MediaTimePoint>
        <MediaDuration>PT0M15S</MediaDuration>
    </MediaTime>
    <VisualDescriptor xsi:type="CameraMotionType">
        <Segment xsi:type="MixtureCameraMotionSegmentType">
            <MediaTime>
                <MediaTimePoint>T00:00:00</MediaTimePoint>
                <MediaDuration>PT0M15S</MediaDuration>
            </MediaTime>
            <FractionalPresence>
                <TrackLeft>5</TrackLeft>
            </FractionalPresence>
            <AmountOfMotion>
                <TrackRight>4</TrackRight>
            </AmountOfMotion>
        </Segment>
    </VisualDescriptor>
</VideoSegment>
<VideoSegment id="id3">
    <MediaTime>
        <MediaTimePoint>T00:00:10</MediaTimePoint>
        <MediaDuration>PT0M10S</MediaDuration>
    </MediaTime>
    <!-- CameraMotion descriptor value omitted -->
</VideoSegment>
<VideoSegment id="id4">
    <MediaTime>
        <MediaTimePoint>T00:00:20</MediaTimePoint>
        <MediaDuration>PT0M10S</MediaDuration>
    </MediaTime>
    <!-- CameraMotion descriptor value omitted -->
</VideoSegment>
</TemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

7.9 Affective description

7.9.1 Introduction

This subclause defines tools for describing affective (emotional) response to multimedia content.

7.9.2 Affective DS

7.9.2.1 Introduction

The Affective DS describes an audience's affective (emotional, mood, or analytical) response to multimedia content. The Affective DS attaches a numeric score to each member of a group of description tool instances. The score measures the relative (within the group of instances) intensity of an audience's affective response to those instances, with respect to a specified affect type (e.g. excitement, happiness, anger, etc).

For example, when scores are assigned to video segments along a story's timeline based on the mood/emotional response of an audience, the resulting description can describe how an audience's mood/emotion changes while viewing the video. A similar approach may be taken to capture the *story shape*—i.e. the shape characterizing the rise and fall in story "complication" as the story progresses. The Affective DS may also be used to describe the

affective response to the semantic entities (events and objects) depicted in video content. For example, the events (goals, passes, and so forth) and objects (players) depicted in the video of a soccer match.

7.9.2.2 Affective DS syntax

```
<!-- ##### -->
<!-- Definition of Affective DS (7.9.2) -->
<!-- ##### -->

<!-- Definition of Affective DS -->
<complexType name="AffectiveType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Type" type="mpeg7:TermUseType"/>
        <element name="Score" minOccurs="2" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="mpeg7:minusOneToOneType">
                <attributeGroup ref="mpeg7:referenceGrp"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

7.9.2.3 Affective DS semantics

Semantics of the AffectiveType:

Name	Definition
AffectiveType	Describes an audience's affective response to multimedia content.
Type	Indicates the type of affective information being measured. The value of Type may be either chosen from a predefined classification scheme (for example, see Classification Schemes for Affective Type in Annex A) or a free text description.
Score	Indicates the relative intensity value of audience's affective information of the referenced instance on a scale from –1 to 1. The minus range (from –1 to 0) is provided to represent the intensity of opposite affective information within a single scale. For example, "excited" and "bored" may be represented in the single score when the maximum intensity of "excited" and "bored" are allocated to 1.0 and –1.0, respectively. The minimum occurrence of two ("2") indicates that because the values are relative, at least two elements with the value shall be instantiated for their comparison.
referenceGrp	Describes a reference to an instance that is assigned an affective score. The affective score represents the relative intensity of audience's affective information with respect to the instance.

7.9.2.4 Affective DS examples (informative)

The following example demonstrates the use of the Affective DS to describe the story shape of a movie. The story shape represents the development of a story along time by assigning a score to each scene, which measures the degree of "story complication" in that scene. The peaks in the score usually correspond to climaxes in the story. Figure 1 shows an example of the story shape characterized for a certain action movie.

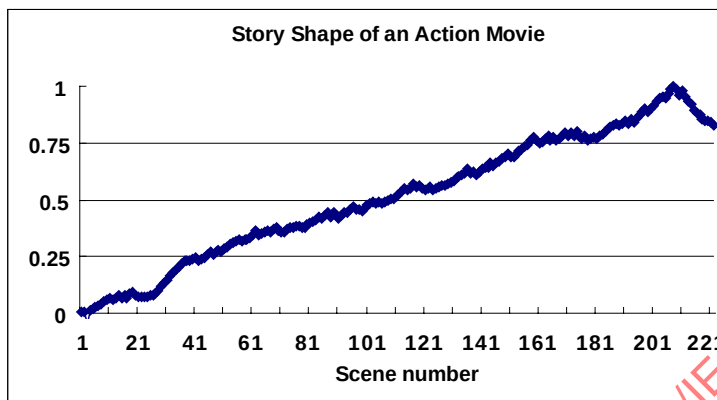


Figure 17 — Illustration of the story shape of an action movie

The description of the story shape shown in Figure 17 is given as follows:

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <!--
      Story Shape description by Affective DS.
      Note: Story shape is defined in the informative classification scheme
      identified by "AffectTypeCS".
    -->
    <Affective id="affective1">
      <Header xsi:type="DescriptionMetadataType">
        <Confidence>0.85</Confidence>
        <Version>1.0</Version>
        <Comment>
          <FreeTextAnnotation>
            Confidence is measured using a certain statistical technique.
            Noticeable experimental conditions are used to describe
            the creation information.
          </FreeTextAnnotation>
        </Comment>
        <PrivateIdentifier>example1</PrivateIdentifier>
        <Creator>
          <Role href="urn:mpeg:cs:AffectBasedContentAnalysisCS:2001">
            <Name>experimentalSubject</Name>
          </Role>
          <Agent xsi:type="OrganizationType">
            <Name>N.Univ.</Name>
          </Agent>
        </Creator>
        <Creator>
          <Role href="urn:mpeg:cs:AffectBasedContentAnalysisCS:2001">
            <Name>organizer</Name>
          </Role>
          <Agent xsi:type="PersonType">
            <Name>
```

```

        <GivenName>Yoshiaki</GivenName>
        <FamilyName>Shibata</FamilyName>
    </Name>
</Agent>
</Creator>
<CreationLocation>
    <Region>jp</Region>
    <AdministrativeUnit>Tokyo</AdministrativeUnit>
</CreationLocation>
<CreationTime>1999-09-12T13:45:00+09:00</CreationTime>
<Instrument>
    <Tool>
        <Name>SemanticScoreMethod</Name>
    </Tool>
</Instrument>
</Header>
<Type href="urn:mpeg:cs:AffectTypeCS:2001">
    <Name>storyShape</Name>
</Type>
<Score idref="scene1">0.00360117</Score>
<Score idref="scene2">0.00720234</Score>
<!-- : -->
<Score idref="scene222">0.825006</Score>
</Affective>
<!--
    Description of AudioVisualSegment Decomposition
-->
<MultimediaContent xsi:type="AudioVisualType">
    <AudioVisual id="program" mediaTimeBase="./MediaLocator"
        mediaTimeUnit="PT1N30F">
        <MediaLocator>
            <MediaUri>http://www.mpeg.org/contents/TheMaskOfZorro.mpg</MediaUri>
        </MediaLocator>
        <!-- 0.0 .. 7669.5 sec for "program" -->
        <MediaTime>
            <MediaRelTimePoint>PT0S</MediaRelTimePoint>
            <!-- <MediaRelTimePoint>PT0S</MediaRelTimePoint> -->
            <MediaDuration>PT2H7M49S5N10F</MediaDuration>
        </MediaTime>
        <!-- Decomposition into scenes -->
        <TemporalDecomposition>
            <AudioVisualSegment id="scene1">
                <!-- 0.0 .. 48.9 sec for "scene1" -->
                <MediaTime>
                    <MediaRelTimePoint>PT0S</MediaRelTimePoint>
                    <MediaDuration>PT48S9N10F</MediaDuration>
                </MediaTime>
            </AudioVisualSegment>
            <AudioVisualSegment id="scene2">
                <!-- 48.9 .. 81.0 sec for "scene2" -->
                <MediaTime>
                    <MediaRelTimePoint>PT48S9N10F</MediaRelTimePoint>
                    <MediaDuration>PT32S1N10F</MediaDuration>
                </MediaTime>
            </AudioVisualSegment>
            <!-- : -->
            <AudioVisualSegment id="scene222">
                <!-- 7639.8 .. 7669.5 sec for "scene222" -->
                <MediaTime>
                    <MediaRelTimePoint>PT2H7M19S8N10F</MediaRelTimePoint>
                    <MediaDuration>PT29S3501N5000F</MediaDuration>

```

```

        </MediaTime>
      </AudioVisualSegment>
    </TemporalDecomposition>
  </AudioVisual>
</MultimediaContent>
</Description>
</Mpeg7>

```

This description consists of two parts:

1. The instance of Affective DS with the DescriptionMetadata DS as a header, where each affective value is associated with a sub-segment (scene) by referencing its identifier ("id") using the idref attribute.
2. The hierarchical decomposition of the movie. The AV segment representing the entire movie ("program") is decomposed into 222 sub-segments called "scenes".

The "storyShape" is specified in the Type element by referencing a term defined in the informative Affective Type Classification Scheme (See Annex A). The information included in the DescriptionMetadata header indicates that the scores were obtained from students in N Univ. based on the Semantic Score Method.

Note: the value of the Confidence element in the Header is 0.85, indicating that the score was obtained using a method that can provide a reliability for the score. In this case, the score is actually a composite score synthesized from the individual score of many individual students using statistical techniques.

Another example of using the Affective DS is to describe the affective response to the events and objects depicted in a soccer game video. Suppose that the video contains a goal event, where a goalkeeper "A" fails to capture a slow ball kicked by a forward "B", resulting in losing a point for "A's" team. The objects, the goalkeeper "A" and the forward "B", may be described using the SemanticPerson DS. Because a fan of "B's" team may feel angrier towards the goalkeeper, who fails to catch such an easy ball, than towards the forward who kicks the ball, his/her affective response to the object descriptions can be represented as:

```

<Affective>
  <Type href="AffectTypeCS:2001">
    <Name> anger </Name>
  </Type>
  <Score idref="Goalkeeper-id">0.8</Score>
  <Score idref="Forward-id">0.2</Score>
</Affective>

```

In this description, "Goalkeeper-id" and "Forward-id" reference the goalkeeper and the forward descriptions respectively, which are assumed to have been described elsewhere.

7.10 Phonetic description

7.10.1 Introduction

This subclause specifies tools for describing various kind of phonetic information.

7.10.2 PhoneticTranscriptionLexicon header

7.10.2.1 Introduction

The PhoneticTranscriptionLexicon header describes a phonetic transcription for a list of words or phrases. It can be viewed as a kind of phonetic transcription dictionary. If a PhoneticTranscriptionLexicon is included in a description, it can be used as a pronunciation dictionary.

7.10.2.2 PhoneticTranscriptionLexicon header syntax

```

<!-- ##### -->
<!-- Definition of PhoneticTranscriptionLexicon header (7.10.2) -->
<!-- ##### -->

<!-- Definition of PhoneticTranscriptionLexicon header -->
<complexType name="PhoneticTranscriptionLexiconType">
  <complexContent>
    <extension base="mpeg7:HeaderType">
      <sequence>
        <element name="Token" minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="mpeg7:WordType">
                <attribute name="phoneticTranscription" use="required">
                  <simpleType>
                    <list itemType="mpeg7:PhoneType"/>
                  </simpleType>
                </attribute>
                <attribute ref="xml:lang" use="optional"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </sequence>
      <attribute ref="xml:lang" use="optional"/>
      <attribute name="phoneticAlphabet" type="mpeg7:phoneticAlphabetType"
        use="optional" default="sampa"/>
    </extension>
  </complexContent>
</complexType>

```

7.10.2.3 PhoneticTranscriptionLexicon header semantics

Name	Definition
PhoneticTranscriptionLexiconType	Describes a lexicon of words and their corresponding phonetic transcriptions.
Token	Describes one word in the lexicon. Each word is coded as a string. WordType is defined in ISO/IEC 15938-4.
phoneticTranscription	Describes the phonetic information about the corresponding word. Phonetic information is represented as a list of phones—a unit of phonetic information. Each phone can be written in a phonetic alphabet, which is specified by the phoneticAlphabet attribute (e.g. the IPA (International Phonetic Alphabet). The PhoneType datatype is defined in ISO/IEC 15938-4.
xml:lang	Indicates the language of the word. This attribute may be used to override a default value for xml:lang specified for this PhoneticTranscriptionLexicon.
phoneticAlphabet	Indicates the phonetic alphabet that is used to write the phoneticTranscription for every Token in this lexicon. The phoneticAlphabetType is defined in ISO/IEC 15938-4.

xml:lang	Specifies the default language of this PhoneticTranscriptionLexicon.

8 Media description tools

8.1 Introduction

This clause specifies tools for describing the media features of the coded multimedia data. The coded multimedia data may be available in multiple modalities, formats, coding versions, or as multiple instances. The tools defined in this clause allow the description of an original instance of coded multimedia data and the multiple variations of the original instance. The following tools are specified in this clause:

Table 12 — Overview of Media Information Tools

<i>Tool</i>	<i>Functionality</i>
Media Information Tools	Describes media features of the coded multimedia data. The MediaInformation DS provides an identifier for the content entity and a set of coding parameters grouped into profiles. The MediaIdentification DS allows the description of the content entity. The different MediaProfile DS instances allow the description of the different sets of coding parameters values available for different coding profiles.

The different instances of the multimedia data can result from capturing a reality with different acquisition devices, or by representing the content using different coding and storage formats, or by coding the data at different rates, qualities, or other coding parameters, and so forth. For example, consider a musical concert which is recorded in two different modalities: audio and AV. Furthermore, the content in each of these *modalities* can be encoded using different methods, which results in multiple *profiles*. In addition, same encoded multimedia content can be represented with multiple physical representations, which results in multiple *instances*. The concepts of modality, profile and instance are illustrated in Figure 18.

- **Content entity:** Each content entity provides a particular form for representing a reality. A single reality, such as a musical concert, can be represented by multiple instances and multiple types of media, e.g. audio, video, images, and so forth.
- **Media information:** The MediaInformation DS describes the media features of the coded multimedia data. A single instance of MediaInformation DS shall be used for each content entity. The MediaInformation DS provides an identifier for each content entity and provides a set of descriptors for describing its media features.
- **Media profile:** A media profile corresponds to the different codings of the entity. Each content entity can have one or more media profiles. The master profile corresponds to the original content entity as initially created or recorded. The other profiles result from transcoding or other coding operations applied to the master. The media profiles may result from encoding the content using different coding parameters for the same encoding algorithms, such as for encoding at different rates, qualities, or resolutions.
- **Media instance:** Media instances represent the different instantiations of the content as physical entities. Each media instance is described using an identifier and a locator.
- **Creation information** (see clause 9): The CreationInformation DS describes information about the process of creating a content entity. The creation information of each content entity can be described using a single instance of the CreationInformation DS.
- **Usage information** (see clause 10): The UsageInformation DS describes the information about usage for a content entity. The usage information of each content entity can be described using a single instance of the UsageInformation DS.

© ISO/IEC 2003 – All rights reserved

The MediaProfile DS is further composed of: (1) an optional MediaFormat D, (2) an optional MediaTranscodingHints D, (3) an optional MediaQuality D, and (4) zero or more MediaInstance DSs. The ComponentMediaProfile DS describes multiplexed media profiles, which correspond to content composed from several multimedia objects or streams, such as media objects in a MPEG-4 stream, or multilingual audio streams and video streams in DVD content, or video streams from multiple cameras in multiplexed video.

8.2 Media information tools

8.2.1 MediaInformation DS

8.2.1.1 Introduction

The MediaInformation DS identifies and describes the media-specific information of the multimedia data.

8.2.1.2 MediaInformation DS syntax

```
<!-- ##### -->
<!-- Definition of MediaInformation DS (8.2.1) -->
<!-- ##### -->

<!-- Definition of MediaInformation DS -->
<complexType name="MediaInformationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="MediaIdentification"
          type="mpeg7:MediaIdentificationType" minOccurs="0"/>
        <element name="MediaProfile" type="mpeg7:MediaProfileType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

8.2.1.3 MediaInformation DS semantics

Semantics of the MediaInformationType:

Name	Definition
MediaInformationType	Describes the physical format of the multimedia data.
MediaIdentification	Identifies the multimedia content entity representing a reality (optional).
MediaProfile	Describes a media profile for the multimedia content.

8.2.2 MediaIdentification D

8.2.2.1 Introduction

The MediaIdentification D identifies the multimedia content independently of the different available instances. The different media profiles of the multimedia content are described via their media profile (see clause 8.2.3), and for each media profile there can be multiple media instances (see clause 8.2.7).

8.2.2.2 MediaIdentification D syntax

```

<!-- ##### -->
<!-- Definition of MediaIdentification D (8.2.2) -->
<!-- ##### -->

<!-- Definition of MediaIdentification D -->
<complexType name="MediaIdentificationType">
  <complexContent>
    <extension base="mpeg7:DType">
      <sequence>
        <element name="EntityIdentifier" type="mpeg7:UniqueIDType"/>
        <element name="AudioDomain" type="mpeg7:ControlledTermUseType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="VideoDomain" type="mpeg7:ControlledTermUseType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="ImageDomain" type="mpeg7:ControlledTermUseType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

8.2.2.3 MediaIdentification D semantics

Semantics of the MediaIdentificationType:

Name	Definition
MediaIdentificationType	Identifies the content entity independently of the different available profiles and instances.
EntityIdentifier	Identifies uniquely the particular and unique multimedia content entity. For example, ISO's ISAN.
AudioDomain	Describes the source, acquisition, and/or use of an audio content entity (optional). If a content entity is AV, AudioDomain and VideoDomain are used simultaneously. An example of CS is AudioDomainCS.
VideoDomain	Describes the source, acquisition, and/or use of a video content entity (optional). An example of CS is VideoDomainCS.
ImageDomain	Describes the source, acquisition, and/or use of an image content entity (optional). An example of CS is ImageDomainCS.

8.2.3 MediaProfile DS

8.2.3.1 Introduction

The MediaProfile DS describes one profile of the multimedia content being described. A profile refers to a variation of the content that is produced from the original or master media by representing the content using different coding parameters, storage format, and so forth. The profile corresponding to the original or master copy of the multimedia content is considered the master media profile. For each profile there can be one or more media instances of the master media profile.

8.2.3.2 MediaProfile DS syntax

```
<!-- ##### -->
<!-- Definition of MediaProfile DS (8.2.3) -->
<!-- ##### -->

<!-- Definition of MediaProfile DS -->
<complexType name="MediaProfileType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="ComponentMediaProfile" type="mpeg7:MediaProfileType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="MediaFormat" type="mpeg7:MediaFormatType"
          minOccurs="0"/>
        <element name="MediaTranscodingHints"
          type="mpeg7:MediaTranscodingHintsType" minOccurs="0"/>
        <element name="MediaQuality" type="mpeg7:MediaQualityType"
          minOccurs="0"/>
        <element name="MediaInstance" type="mpeg7:MediaInstanceType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="master" type="boolean" use="optional" default="false"/>
    </extension>
  </complexContent>
</complexType>
```

8.2.3.3 MediaProfile DS semantics

Semantics of the MediaProfileType:

Name	Definition
MediaProfileType	Describes the media profile of the multimedia content being described.
ComponentMediaProfile	Describes one component Media Profile of a multiplexed Media Profile, describing a multimedia content with different component multimedia contents (optional).
MediaFormat	Describes the format of the Media Profile (optional).
MediaTranscodingHints	Describes the transcoding hints of the Media Profile (optional).
MediaQuality	Describes the quality of the signal corresponding to the Media Profile (optional).
MediaInstance	Identifies and locates one Media Instance of the Media Profile (optional).
master	Indicates whether the Media Profile corresponds to the original or master profile of the multimedia content entity. Default value of this attribute is <i>false</i> .

The ComponentMediaProfile DS within the MediaProfile DS allows the description of multiplexed content streams. Typical examples of multiplexed streams include:

- MPEG-4 streams that are composed from multiple media objects,
- DVD content in which multiple audio streams in different languages are multiplexed and synchronized with the video,
- A globally synchronized multimedia content stream composed from multiple audio and video streams acquired from different microphones and cameras, which may provide different views of the same event.

8.2.4 MediaFormat D

8.2.4.1 Introduction

The MediaFormat D describes the information related to the file format and the coding parameters of the media profile.

8.2.4.2 MediaFormat D syntax

```
<!-- ##### -->
<!-- Definition of MediaFormat D (8.2.4) -->
<!-- ##### -->

<!-- Definition of MediaFormat D -->
<complexType name="MediaFormatType">
  <complexContent>
    <extension base="mpeg7:DType">
      <sequence>
        <element name="Content" type="mpeg7:ControlledTermUseType"/>
        <element name="Medium" type="mpeg7:ControlledTermUseType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

<element name="FileFormat" type="mpeg7:ControlledTermUseType"
  minOccurs="0"/>
<element name="FileSize" type="nonNegativeInteger" minOccurs="0"/>
<element name="System" type="mpeg7:ControlledTermUseType"
  minOccurs="0"/>
<element name="Bandwidth" type="float" minOccurs="0"/>
<element name="BitRate" minOccurs="0">
  <complexType>
    <simpleContent>
      <extension base="nonNegativeInteger">
        <attribute name="variable" type="boolean"
          use="optional" default="false"/>
        <attribute name="minimum" type="nonNegativeInteger"
          use="optional"/>
        <attribute name="average" type="nonNegativeInteger"
          use="optional"/>
        <attribute name="maximum" type="nonNegativeInteger"
          use="optional"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="TargetChannelBitRate" type="nonNegativeInteger"
  minOccurs="0"/>
<element name="ScalableCoding" minOccurs="0">
  <simpleType>
    <union>
      <simpleType>
        <restriction base="NMTOKEN">
          <enumeration value="spatial"/>
          <enumeration value="temporal"/>
          <enumeration value="snr"/>
          <enumeration value="fgs"/>
        </restriction>
      </simpleType>
      <simpleType>
        <restriction base="mpeg7:termReferenceType"/>
      </simpleType>
    </union>
  </simpleType>
</element>
<element name="VisualCoding" minOccurs="0">
  <complexType>
    <sequence>
      <element name="Format" minOccurs="0">
        <complexType>
          <complexContent>
            <extension base="mpeg7:ControlledTermUseType">
              <attribute name="colorDomain" use="optional"
                default="color">
                <simpleType>
                  <union>
                    <simpleType>
                      <restriction base="NMTOKEN">
                        <enumeration value="binary"/>
                        <enumeration value="color"/>
                        <enumeration
                          value="graylevel"/>
                        <enumeration
                          value="colorized"/>

```



```

        </restriction>
      </simpleType>
      <simpleType>
        <restriction
          base="mpeg7:termReferenceType"/>
      </simpleType>
    </union>
  </simpleType>
</attribute>
</extension>
</complexContent>
</complexType>
</element>
<element name="Pixel" minOccurs="0">
  <complexType>
    <attribute name="resolution" type="nonNegativeInteger"
      use="optional"/>
    <attribute name="aspectRatio"
      type="mpeg7:nonNegativeReal"
      use="optional"/>
    <attribute name="bitsPer" type="nonNegativeInteger"
      use="optional"/>
  </complexType>
</element>
<element name="Frame" minOccurs="0">
  <complexType>
    <attribute name="height" type="nonNegativeInteger"
      use="optional"/>
    <attribute name="width" type="nonNegativeInteger"
      use="optional"/>
    <attribute name="aspectRatio"
      type="mpeg7:nonNegativeReal"
      use="optional"/>
    <attribute name="rate" type="mpeg7:nonNegativeReal"
      use="optional"/>
    <attribute name="structure" use="optional">
      <simpleType>
        <restriction base="NMTOKEN">
          <enumeration value="progressive"/>
          <enumeration value="interlaced"/>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>
</element>
<element name="ColorSampling" type="mpeg7:ColorSamplingType"
  minOccurs="0"/>
</sequence>
</complexType>
</element>
<element name="AudioCoding" minOccurs="0">
  <complexType>
    <sequence>
      <element name="Format" type="mpeg7:ControlledTermUseType"
        minOccurs="0"/>
      <element name="AudioChannels" minOccurs="0">
        <complexType>
          <simpleContent>
            <extension base="nonNegativeInteger">
              <attribute name="front"

```

```

        type="nonNegativeInteger" use="optional"/>
        <attribute name="side" type="nonNegativeInteger"
        use="optional"/>
        <attribute name="rear" type="nonNegativeInteger"
        use="optional"/>
        <attribute name="lfe" type="nonNegativeInteger"
        use="optional"/>
        <attribute name="track"
        type="nonNegativeInteger" use="optional"/>
    </extension>
</simpleContent>
</complexType>
</element>
<element name="Sample" minOccurs="0">
    <complexType>
        <attribute name="rate" type="mpeg7:nonNegativeReal"
        use="optional"/>
        <attribute name="bitsPer" type="nonNegativeInteger"
        use="optional"/>
    </complexType>
</element>
<element name="Emphasis" minOccurs="0">
    <simpleType>
        <union>
            <simpleType>
                <restriction base="NMTOKEN">
                    <enumeration value="none"/>
                    <enumeration value="50over15Microseconds"/>
                    <enumeration value="ccittJ17"/>
                </restriction>
            </simpleType>
            <simpleType>
                <restriction base="mpeg7:termReferenceType"/>
            </simpleType>
        </union>
    </simpleType>
</element>
<element name="Presentation"
        type="mpeg7:ControlledTermUseType" minOccurs="0"/>
</sequence>
</complexType>
</element>
<element name="SceneCodingFormat" type="mpeg7:ControlledTermUseType"
minOccurs="0"/>
<element name="GraphicsCodingFormat" type="mpeg7:ControlledTermUseType"
minOccurs="0"/>
<element name="OtherCodingFormat" type="mpeg7:ControlledTermUseType"
minOccurs="0"/>
</sequence>
</extension>
</complexContent>
</complexType>

<!-- Definition of ColorSampling D -->
<complexType name="ColorSamplingType">
    <complexContent>
        <extension base="mpeg7:DType">
            <sequence>
                <element name="Lattice" minOccurs="0">
                    <complexType>

```

```

        <attribute name="width" type="positiveInteger" use="required"/>
        <attribute name="height" type="positiveInteger" use="required"/>
    </complexType>
</element>
<element name="Field" minOccurs="1" maxOccurs="unbounded">
    <complexType>
        <sequence>
            <element name="Component" minOccurs="1"
                maxOccurs="unbounded">
                <complexType>
                    <complexContent>
                        <extension base="mpeg7:TermUseType">
                            <sequence>
                                <element name="Offset" minOccurs="0">
                                    <complexType>
                                        <attribute name="horizontal"
                                            type="mpeg7:nonNegativeReal"
                                            use="required"/>
                                        <attribute name="vertical"
                                            type="mpeg7:nonNegativeReal"
                                            use="required"/>
                                    </complexType>
                                </element>
                                <element name="Period" minOccurs="0">
                                    <complexType>
                                        <attribute name="horizontal"
                                            type="mpeg7:nonNegativeReal"
                                            use="required"/>
                                        <attribute name="vertical"
                                            type="mpeg7:nonNegativeReal"
                                            use="required"/>
                                    </complexType>
                                </element>
                            </sequence>
                        </extension>
                    </complexContent>
                </complexType>
            </element>
        </sequence>
        <attribute name="temporalOrder"
            type="nonNegativeInteger"
            use="optional"/>
        <attribute name="positionalOrder"
            type="nonNegativeInteger"
            use="optional"/>
    </complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

```

8.2.4.3 MediaFormat D semantics

Semantics of the MediaFormatType:

Name	Definition
MediaFormatType	Describes the format and coding parameters of the Media Profile.
Content	Describes the media present in the Media Profile (e.g., audio, image, scene definition, video, audiovisual). An example of CS is ContentCS.
Medium	Describes the physical storage medium on which the Media Profile is stored (optional). An example of CS is MediumCS.
FileFormat	Describes the file format of the Media Profile (optional). An example of CS is FileFormatCS.
FileSize	Indicates the size, in bytes, of the file where the Media Profile is stored (optional).
System	Describes the broad media format of the Media Profile (optional). An example of CS is SystemCS.
Bandwidth	Indicates the bandwidth range in Hz covered by the coded multimedia content (optional). Its value depends on the acquisition filters or transcoding applied to the Media Profile.
BitRate	Indicates the nominal bit rate in bits/s of the Media Profile (optional).
variable	Indicates whether the bitrate is variable or fixed. If the bitrate is variable, three optional attributes can be used to specify the minimum, maximum and average bitrates. Default value of this attribute is <i>false</i> .
minimum	Indicates the minimum numerical value for the BitRate in case of variable bit rate (optional).
maximum	Indicates the maximum numerical value for the BitRate in case of variable bit rate (optional).
average	Indicates the average numerical value for the BitRate in case of variable bit rate (optional).
TargetChannelBitRate	Indicates the nominal bit rate in bits/s of the channel for which this MediaProfile is targeted (optional).
ScalableCoding	Indicates the scalability used in the coding (optional). The values of the scalable coding are defined as follows: • <i>spatial</i> – The scalability is spatial. • <i>temporal</i> – The scalability is temporal. • <i>snr</i> – The scalability is SNR (Signal to Noise Ratio). • <i>fgs</i> – The scalability is FGS (Fine Granularity Scalability). Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.

Name	Definition
VisualCoding	Describes the coding of the visual component of the Media Profile (optional). If a content entity is AV, VisualCoding and AudioCoding are used simultaneously.
Format	Describes the coding format of the visual component of the Media Profile (optional). If a content entity is AV, VisualCoding and AudioCoding are used simultaneously. An example of CS is VisualCodingFormatCS.
colorDomain	Describes the color domain of the image or video. The values of the color domain are defined as follows. • <i>binary</i> — The image or video is binary. • <i>color</i> — The image or video is color. This is the default value. • <i>graylevel</i> — The image or video is graylevel. • <i>colorized</i> — The image or the video was originally a graylevel and was given color afterwards. Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.
Pixel	Describes the pixels of images and video frames (optional).
resolution	Indicates the resolution of images and frames in dpi (dots per inch) (optional).
aspectRatio	Describes the pixel aspect ratio (width/height) (optional).
bitsPer	Indicates the visual sample (pixel) accuracy in bits per pixel (optional).
Frame	Describes the frame of images and video frames.
height	Indicates the height of images and frames in pixels. The quotient of width and height is the sampling aspect ratio, which is the same as the optical image aspect ratio if the pixels are square (optional).
width	Indicates the width of images and frames in pixels. The quotient of width and height is the sampling aspect ratio, which is the same as the optical image aspect ratio if the pixels are square (optional).
aspectRatio	Describes the optical image aspect ratio (width/height). It is the quotient of width and height in non-pixel based presentation units (e.g. screen dimensions for films). It is the same as the sampling aspect ratio if the pixels are square (optional).
rate	Indicates the frame rate in Hz (optional).
structure	Describes the temporal sampling structure used for the coding of the video (optional). The values of the temporal sampling structure are defined as follows: • <i>progressive</i> — The sampling structure is progressive (non-interlaced). • <i>interlaced</i> — The sampling structure is interlaced (2:1, two fields interlaced in each frame).
ColorSampling	Describes the color sampling on the video component of the Media Profile (optional).

Name	Definition
AudioCoding	Describes the coding of the audio component of the Media Profile (optional). If a content entity is AV, VisualCoding and AudioCoding are used simultaneously.
Format	Describes the coding format of the audio component of the Media Profile (optional). If a content entity is AV, VisualCoding and AudioCoding are used simultaneously. An example of CS is AudioCodingFormatCS.
AudioChannels	Describes the number and optionally the configuration of audio channels in the Media Profile (optional). The channel configuration information can be described by use of 'front', 'side', 'rear' and 'lfe' attributes. In case of multilingual or multitrack audio stream, the number of tracks can be described on 'track' attribute.
front	Indicates the number of the front channels (optional).
side	Indicates the number of the side channels (optional).
rear	Indicates the number of the rear channels (optional).
lfe	Indicates the number of the LFE (Low Frequency Enhancement) channels (optional).
track	Indicates the number of independent audio tracks used in a multilingual or multitrack audio stream (optional).
Sample	Describes the audio samples (optional).
rate	Indicates the sampling rate for audio in Hz (optional).
bitsPer	Indicates the audio sample accuracy in bits per sample (optional).
Emphasis	Indicates the audio pre-emphasis used in the encoding of the multimedia content (optional). The values of the emphasis are defined as follows: • <i>none</i> – Audio emphasis is not used. • <i>50over15Microseconds</i> – Audio emphasis is 50/15 microseconds. • <i>ccittJ17</i> – Audio emphasis is CCITT J.17. Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.
Presentation	Describes the audio presentation format of the Media Profile (optional). An example of CS is AudioPresentationCS. AudioChannels allows to define channel configuration in a more detailed way.
SceneCodingFormat	Describes the coding format for a scene composition stream (optional), for example a BIFS stream in MPEG-4. An example of CS is SceneCodingFormatCS.
GraphicsCodingFormat	Describes the coding format for a graphics stream (optional), for example VRML. An example of CS is GraphicsCodingFormatCS.
OtherCodingFormat	Describes other coding formats that are not visual, audio, scene description or graphics such as HTML or Flash (optional). An example of CS is OtherCodingFormatCS.

Semantics of the ColorSamplingType:

Name	Definition
ColorSamplingType	Describes the color sampling of a video given any number of color components and any number of temporally sited fields.
Lattice	Describes the imaginary lattice the sub-sampling is derived from (optional).
width	Indicates the width in pixels (all components combined) of the lattice in units of D_h . D_h represents the horizontal distance between pixels in the lattice (see Figure 19).
height	Indicates the height in pixels (all components combined) of the lattice in units of D_v . D_v represents the vertical distance between pixels in the lattice (see Figure 19).
Field	Describes each field and each color component information on a per field basis, for multiple temporally fields based videos.
Component	Describes each color component of the video.
Offset	Describes the offset in the lattice for the color component (optional).
horizontal	Indicates the horizontal offset (in units of D_h) from (0,0) in the lattice of the first sample for this component
vertical	Indicates the vertical offset (in units of D_v) from (0,0) in the lattice of the first sample for this component
Period	Describes the repeating distance in the lattice for the color component (optional).
horizontal	Indicates the repeating horizontal distance (in units of D_h) from one sample in the lattice to the next closest neighbor of this component in the lattice.
vertical	Indicates the repeating vertical distance (in units of D_v) from one sample in the lattice to the next closest neighbor of this component in the lattice. This distance takes into account the temporal displacement from field to field as well.
temporalOrder	Indicates the order in time that this field is displayed. 0 being first, 1 being second (optional).
positionalOrder	Indicates the order in space that this field is displayed. 0 would be the "top" most field in an interlaced frame, 1 would be the bottom most field (optional).

The ColorSamplingType characterizes the color sampling used in the media. A generic way of describing the sampling used is provided to describe both a sub-sampling in a line of pixels but also the siting of those pixels. It also takes into account a temporal siting of the pixels as differing media can be sampled at differing frequencies. Finally, it takes into account any number of components in a pixel.

The ColorSampling D bases values on an imaginary lattice with D_v representing the vertical distance between pixels and D_h representing the horizontal distance between pixels. O_h and O_v represent starting offsets for both horizontal and vertical distance from the top left pixel in the lattice. P_v and P_h represent the period of repeating components in units of D_v and D_h .

Figure 19 shows the variables that can be specified on a per component basis.

Note: two fields that vary temporally are superimposed on one another. In the example the distance from the first luminance sample in the first row in field 1 to the first luminance sample in the second row of field 1 is $2D_v$.

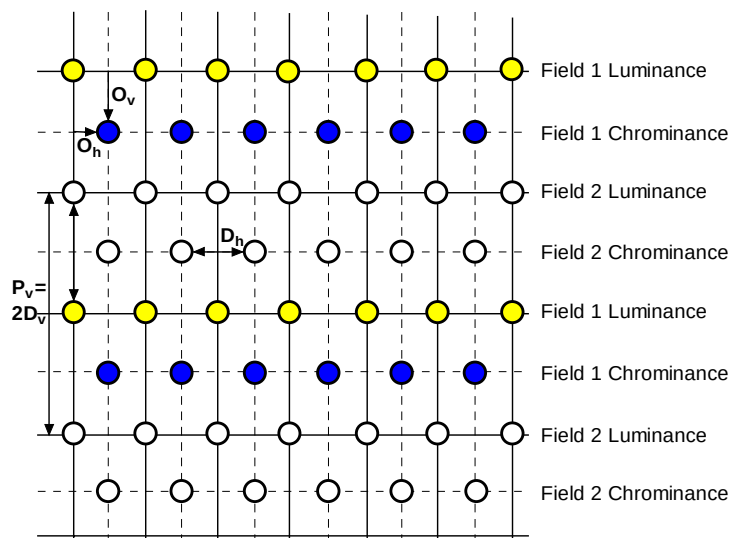


Figure 19 — Illustration of color sampling values on an imaginary lattice

8.2.5 MediaTranscodingHints D

8.2.5.1 Introduction

The MediaTranscodingHints D specifies transcoding hints for the media. The purpose of this descriptor is to improve the quality of transcoded content and to reduce the complexity of transcoding operations.

8.2.5.2 MediaTranscodingHints D syntax

```
<!-- ##### -->
<!-- Definition of Media TranscodingHints D (8.2.5) -->
<!-- ##### -->

<!-- Definition of MediaTranscodingHints D -->
<complexType name="MediaTranscodingHintsType">
  <complexContent>
    <extension base="mpeg7:DType">
      <sequence>
        <element name="MotionHint" minOccurs="0">
          <complexType>
            <sequence>
              <element name="MotionRange" minOccurs="0">
                <complexType>
                  <attribute name="xLeft"
                    type="nonNegativeInteger" use="required"/>
                  <attribute name="xRight"
                    type="nonNegativeInteger" use="required"/>
                  <attribute name="yDown"
                    type="nonNegativeInteger" use="required"/>
                  <attribute name="yUp"
                    type="nonNegativeInteger" use="required"/>
                </complexType>
              </element>
            </sequence>
          </complexType>
        </element>
        <attribute name="uncompensability" type="mpeg7:zeroToOneType"
          use="optional"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```



```

        <attribute name="intensity" type="mpeg7:zeroToOneType"
            use="optional"/>
    </complexType>
</element>
<element name="ShapeHint" minOccurs="0">
    <complexType>
        <attribute name="shapeChange" type="float" use="optional"/>
        <attribute name="numOfNonTranspBlocks"
            type="mpeg7:nonNegativeReal" use="optional"/>
    </complexType>
</element>
<element name="CodingHints" minOccurs="0">
    <complexType>
        <attribute name="avgQuantScale" type="mpeg7:nonNegativeReal"
            use="optional"/>
        <attribute name="intraFrameDistance" type="nonNegativeInteger"
            use="optional"/>
        <attribute name="anchorFrameDistance"
            type="positiveInteger" use="optional"/>
    </complexType>
</element>
</sequence>
<attribute name="difficulty" type="mpeg7:zeroToOneType" use="optional"/>
<attribute name="importance" type="mpeg7:zeroToOneType" use="optional"/>
<attribute name="spatialResolutionHint" type="mpeg7:zeroToOneType"
    use="optional"/>
</extension>
</complexContent>
</complexType>

```

8.2.5.3 MediaTranscodingHints D semantics

Semantics of the MediaTranscodingHintsType:

Name	Definition
MediaTranscodingHintsType	Describes transcoding hints of the Media Profile.
MotionHint	Describes the motion hints for a transcoder (optional).
MotionRange	Describe motion range for a transcoder (optional).
xLeft	Indicates the recommended integer pel search range for horizontal motion vectors to the left.
xRight	Indicates the recommended integer pel search range for horizontal motion vectors to the right.
yDown	Indicates the recommended integer pel search range for vertical motion vectors to the bottom.
yUp	Indicates the recommended integer pel search range for vertical motion vectors to the top.
uncompensability	Describes the amount of new content, that is content that cannot be predicted, in the corresponding segment (this descriptor applies to descriptions attached to video segments). The uncompensability takes values from 0.0 to 1.0, where 0.0 indicates no new content and 1.0 indicates

Name	Definition
	the highest change in content. 1.0 means that nothing can be predicted and a transcoder/encoder may choose a non-predictive encoding mode (optional).
intensity	Describes the motion intensity in a segment. The motionIntensity takes values from 0.0 to 1.0, where 0.0 indicates low motion intensity and 1.0 indicates the highest motion intensity (optional).
ShapeHint	Shape hints for the transcoder (optional).
shapeChange	Describes the amount of shape change in the corresponding segment (this descriptor applies to descriptions attached to moving regions). The shapeChange takes values from 0.0 to 1.0, where 0.0 indicates that no change has occurred and 1.0 indicates that all the pixels that define an object have been displaced (optional).
numOfNonTranspBlocks	Describes the average number of 16x16 blocks per frame containing at least one pixel with a non-zero alpha-map value (optional).
CodingHints	Coding hints for the transcoder (optional).
avgQuantScale	Describes the average quantization scale used to compress the media according to the compression format applied (optional).
intraFrameDistance	Describes the distance between Intra-coded Frames, also known as N (optional). A value of N=0 represents the case when N is infinite, for example, when the GOP has no I-frame (PBBPBBP..) or when there is only one I-frame at the start (IPPP..)
anchorFrameDistance	Describes the distance between anchor frames, also known as M, where an anchor frame is defined as a frame that predictions are made from, for example, an I or P frame (optional).
difficulty	Describes the transcoding difficulty of the media in a segment. The difficultyHint takes values from 0.0 to 1.0, where 0.0 indicates the lowest difficulty and 1.0 indicates the highest difficulty. The difficulty is normalized within the contents. The maximum value of the difficulty is 1.0, which indicates the most difficult part to encode within the contents (optional).
importance	Describes the importance with respect to the semantic content of the media. The importance takes values from 0.0 to 1.0, where 0.0 indicates the lowest importance and 1.0 indicates the highest importance (optional).
spatialResolutionHint	Describes the maximum allowable spatial resolution reduction factor for perceptibility. The SpatialResolutionHint takes values from 0.0 to 1.0, where 0.5 indicates that the resolution can be reduced by half and 1.0 indicates the resolution cannot be reduced (optional).

8.2.6 Media Quality D

8.2.6.1 Introduction

The MediaQuality D describes quality rating information of multimedia content. The quality of AV content may decrease when the signal is compressed, transmitted, or converted. The MediaQuality D describes both subjective and objective quality ratings.

8.2.6.2 MediaQuality D syntax

```

<!-- ##### -->
<!-- Definition of MediaQuality D (8.2.6) -->
<!-- ##### -->

<!-- Definition of MediaQuality D -->
<complexType name="MediaQualityType">
  <complexContent>
    <extension base="mpeg7:DType">
      <sequence>
        <element name="QualityRating" minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:RatingType">
                <attribute name="type" use="required">
                  <simpleType>
                    <restriction base="NMTOKEN">
                      <enumeration value="subjective"/>
                      <enumeration value="objective"/>
                    </restriction>
                  </simpleType>
                </attribute>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="RatingSource" type="mpeg7:AgentType"
          minOccurs="0"/>
        <element name="RatingInformationLocator" type="mpeg7:ReferenceType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="PerceptibleDefects" minOccurs="0">
          <complexType>
            <sequence>
              <element name="VisualDefects"
                type="mpeg7:ControlledTermUseType"
                minOccurs="0" maxOccurs="unbounded"/>
              <element name="AudioDefects"
                type="mpeg7:ControlledTermUseType"
                minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of Rating datatype -->
<complexType name="RatingType">
  <sequence>
    <element name="RatingValue" type="float"/>
    <element name="RatingScheme">
      <complexType>
        <complexContent>
          <extension base="mpeg7:TermUseType">
            <attribute name="best" type="float" use="optional"/>
            <attribute name="worst" type="float" use="optional"/>
            <attribute name="style" use="required">
              <simpleType>

```

```

        <restriction base="NMTOKEN">
            <enumeration value="higherBetter"/>
            <enumeration value="lowerBetter"/>
        </restriction>
    </simpleType>
</attribute>
</extension>
</complexContent>
</complexType>
</element>
</sequence>
</complexType>

```

8.2.6.3 MediaQuality D semantics

Semantics of the MediaQualityType:

Name	Definition
MediaQualityType	Describes the media quality of the Media Profile.
QualityRating	Describes the rating values and the criterion used to create the media quality ratings.
type	Describes the type of media quality rating. The types of quality rating are defined as follows. <i>subjective</i> – The rating is subjective, that is, it is provided by human viewers. <i>objective</i> – The rating is objective, that is, it is acquired using computational means.
RatingSource	Describes the source that provides the ratings (optional).
RatingInformationLocator	Indicates the locator for additional information about the quality rating method (optional).
PerceptibleDefects	Describes defects that are perceived in the media (optional).
VisualDefects	Describes the visual errors perceived in the media (optional). Errors are listed in the order of descending severity. An example of CS is VisualDefectsCS.
AudioDefects	Describes the audio errors perceived in the media (optional). Errors are listed in the order of descending severity. An example of CS is AudioDefectsCS.

Semantics of the RatingType:

Name	Definition
RatingType	Describes the rating values and the criterion used to create the ratings. It is used for media quality rating and media review descriptions.
RatingValue	Indicates the actual numerical rating values assigned to the multimedia content.

Name	Definition
RatingScheme	Describes the name of the rating method used to obtain the rating. An example of CS for media quality rating is QualityRatingSchemeCS.
best	Indicates the numerical rating value that means the best rating (optional).
worst	Indicates the numerical rating value that means the worst rating (optional).
style	Describes the relationship between the numerical rating values and rating style. The types of rating styles are defined as follows. <i>higherBetter</i> – The rating style indicates that a higher value means better. <i>lowerBetter</i> – The rating style indicates that a lower value means better.

8.2.7 MediaInstance DS

8.2.7.1 Introduction

The MediaInstance DS identifies and locates the different media instances of a media profile.

8.2.7.2 MediaInstance DS syntax

```

<!-- ##### -->
<!-- Definition of MediaInstance DS (8.2.7) -->
<!-- ##### -->

<!-- Definition of MediaInstance DS -->
<complexType name="MediaInstanceType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="InstanceIdentifier" type="mpeg7:UniqueIDType"/>
        <choice>
          <element name="MediaLocator" type="mpeg7:MediaLocatorType"/>
          <element name="LocationDescription" type="mpeg7:TextualType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

8.2.7.3 MediaInstance DS semantics

Semantics of the MediaInstanceType:

Name	Definition
MediaInstanceType	Describes the identification and the location of Media Instances.
InstanceIdentifier	Identifies the Media Instance or copy.

Name	Definition
MediaLocator	Describes the Media Locator of a Media Instance.
LocationDescription	Describes the location of a Media Instance not available via a Media Locator. For example, the location of tapes stored in the analogue archive of a broadcaster.

8.2.8 Media information examples (informative)

8.2.8.1 MediaInformation DS examples

The following example shows the use of MediaInformation DS for describing the media information for a video. The content is identified as corresponding to "natural" content from the video domain. A single media profile is described, which identifies the medium as "CD", the visual coding format as "MPEG-1 Video", the audio coding format as "MPEG-1 Audio." Furthermore, two instances of the content are described, one belonging to a CD collection, and another provided on-line.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <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>
              <AudioCoding>
                <Format
                  href="urn:mpeg:mpeg7:cs:AudioCodingFormatCS:2001:3">
                  <Name xml:lang="en">MPEG-1 Audio</Name>
                </Format>
                <AudioChannels front="3" rear="2" lfe="1">6
              </AudioChannels>
            </MediaProfile>
          </MediaInformation>
        </Video>
      </MultimediaContent>
    </Description>
  </Mpeg7>
```

```

        </AudioCoding>
    </MediaFormat>
    <MediaTranscodingHints difficulty="0.562789">
        <MotionHint>
            <MotionRange xLeft="15" xRight="15" yDown="15" yUp="15"/>
        </MotionHint>
    </MediaTranscodingHints>
    <MediaQuality>
        <QualityRating type="objective">
            <RatingValue>40.05</RatingValue>
            <RatingScheme style="higherBetter"
                href="urn:mpeg:mpeg7:cs:QualityRatingSchemeCS:2001:2.3">
                <Name xml:lang="en">PSNR_Y</Name>
            </RatingScheme>
        </QualityRating>
        <RatingInformationLocator
            href="Measurement/PicQuality/quality.html"/>
        <PerceptibleDefects>
            <VisualDefects
                href="urn:mpeg:mpeg7:cs:VisualDefectsCS:2001:3">
                <Name xml:lang="en">Edge noise</Name>
            </VisualDefects>
        </PerceptibleDefects>
    </MediaQuality>
    <MediaInstance id="mp7cs17news1">
        <InstanceIdentifier organization="MPEG"
            type="MPEG7ContentSetId"> mpeg7_17/news1
        </InstanceIdentifier>
        <LocationDescription>news1.mpg in Content Set CD 17
        </LocationDescription>
    </MediaInstance>
    <MediaInstance id="onlinemp7cs17news1">
        <InstanceIdentifier organization="MPEG"
            type="MPEG7ContentSetOnlineId"> mpeg7/content17/news1
        </InstanceIdentifier>
        <MediaLocator>
            <MediaUri>
                http://www.mpeg.org/mpeg7/contentset/17/news1.mpg
            </MediaUri>
        </MediaLocator>
    </MediaInstance>
    </MediaProfile>
    </MediaInformation>
    </Video>
    </MultimediaContent>
    </Description>
</Mpeg7>

```

8.2.8.2 ColorSampling D examples

The following example shows the use of the ColorSampling D for describing the following progressive sampling: YCbCr 4:4:4, where all samples co-sited.

```

<Mpeg7>
    <Description xsi:type="ContentEntityType">
        <MultimediaContent xsi:type="VideoType">
            <Video>
                <MediaInformation>
                    <MediaIdentification>

```

```

        <EntityIdentifier organization="MPEG" type="MPEG7ContentSetId">
            mpeg7_content:news1
        </EntityIdentifier>
    </MediaIdentification>
    <MediaProfile>
        <MediaFormat>
            <Content href="MPEG7ContentCS">
                <Name>audiovisual</Name>
            </Content>
            <VisualCoding>
                <ColorSampling>
                    <Lattice width="720" height="486"/>
                    <Field temporalOrder="0" positionalOrder="0">
                        <Component>
                            <Name>Luminance</Name>
                            <Offset horizontal="0.0" vertical="0.0"/>
                            <Period horizontal="1.0" vertical="1.0"/>
                        </Component>
                        <Component>
                            <Name>ChrominanceBlueDifference</Name>
                            <Offset horizontal="0.0" vertical="0.0"/>
                            <Period horizontal="1.0" vertical="1.0"/>
                        </Component>
                        <Component>
                            <Name>ChrominanceRedDifference</Name>
                            <Offset horizontal="0.0" vertical="0.0"/>
                            <Period horizontal="1.0" vertical="1.0"/>
                        </Component>
                    </Field>
                </ColorSampling>
            </VisualCoding>
        </MediaFormat>
    </MediaProfile>
</MediaInformation>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

The following example shows the use of the ColorSampling D for describing the following MPEG-2 interlaced sampling: YCbCr 4:2:2, which is illustrated in Figure 20.

```

<Mpeg7>
    <Description xsi:type="ContentEntityType">
        <MultimediaContent xsi:type="VideoType">
            <Video>
                <MediaInformation>
                    <MediaIdentification>
                        <EntityIdentifier organization="MPEG" type="MPEG7ContentSetId">
                            mpeg7_content:news1
                        </EntityIdentifier>
                    </MediaIdentification>
                    <MediaProfile>
                        <MediaFormat>
                            <Content href="MPEG7ContentCS:2001">
                                <Name>audiovisual</Name>
                            </Content>
                            <VisualCoding>
                                <ColorSampling>

```



```

<Lattice width="720" height="486"/>
<Field temporalOrder="0" positionalOrder="0">
  <Component>
    <Name>Luminance</Name>
    <Offset horizontal="0.0" vertical="0.0"/>
    <Period horizontal="1.0" vertical="2.0"/>
  </Component>
  <Component>
    <Name>ChrominanceBlueDifference</Name>
    <Offset horizontal="0.0" vertical="0.0"/>
    <Period horizontal="2.0" vertical="2.0"/>
  </Component>
  <Component>
    <Name>ChrominanceRedDifference</Name>
    <Offset horizontal="0.0" vertical="0.0"/>
    <Period horizontal="2.0" vertical="2.0"/>
  </Component>
</Field>
<Field temporalOrder="1" positionalOrder="1">
  <Component>
    <Name>Luminance</Name>
    <Offset horizontal="0.0" vertical="1.0"/>
    <Period horizontal="1.0" vertical="2.0"/>
  </Component>
  <Component>
    <Name>ChrominanceBlueDifference</Name>
    <Offset horizontal="0.0" vertical="1.0"/>
    <Period horizontal="2.0" vertical="2.0"/>
  </Component>
  <Component>
    <Name>ChrominanceRedDifference</Name>
    <Offset horizontal="0.0" vertical="1.0"/>
    <Period horizontal="2.0" vertical="2.0"/>
  </Component>
</Field>
</ColorSampling>
</VisualCoding>
</MediaFormat>
</MediaProfile>
</MediaInformation>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

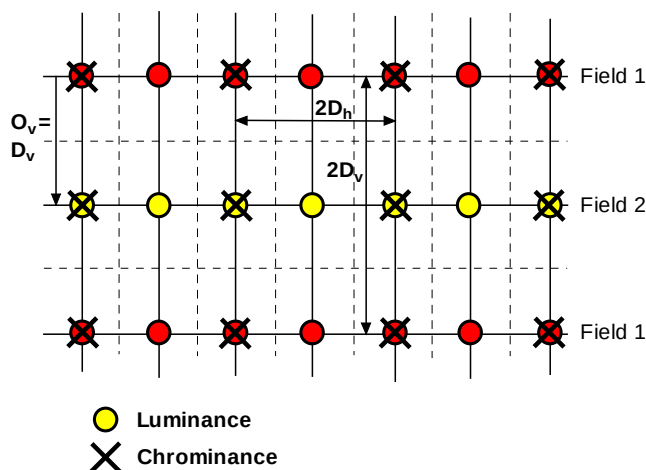


Figure 20 — Illustration of the siting of luminance and chrominance samples for 4:2:2 data (i.e., MPEG-2 interlaced color sampling YCbCr 4:2:2)

The following example shows the use of the ColorSampling D for describing the following MPEG-2 interlaced sampling: YCbCr 4:2:0, which is illustrated in Figure 21.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <MediaInformation>
          <MediaIdentification>
            <EntityIdentifier organization="MPEG" type="MPEG7ContentSetId">
              mpeg7_content:news1
            </EntityIdentifier>
          </MediaIdentification>
          <MediaProfile>
            <MediaFormat>
              <Content href="MPEG7ContentCS:2001">
                <Name>audiovisual</Name>
              </Content>
              <VisualCoding>
                <ColorSampling>
                  <Lattice height="720" width="486"/>
                  <Field temporalOrder="0" positionalOrder="0">
                    <Component>
                      <Name>Luminance</Name>
                      <Offset horizontal="0.0" vertical="0.0"/>
                      <Period horizontal="1.0" vertical="2.0"/>
                    </Component>
                    <Component>
                      <Name>ChrominanceBlueDifference</Name>
                      <Offset horizontal="0.0" vertical="0.5"/>
                      <Period horizontal="2.0" vertical="4.0"/>
                    </Component>
                    <Component>
                      <Name>ChrominanceRedDifference</Name>
                      <Offset horizontal="0.0" vertical="0.5"/>
                      <Period horizontal="2.0" vertical="4.0"/>
                    </Component>
                  </Field>
                  <Field temporalOrder="1" positionalOrder="1">
```

```

<Component>
  <Name>Luminance</Name>
  <Offset horizontal="0.0" vertical="1.0"/>
  <Period horizontal="1.0" vertical="2.0"/>
</Component>
<Component>
  <Name>ChrominanceBlueDifference</Name>
  <Offset horizontal="0.0" vertical="2.5"/>
  <Period horizontal="2.0" vertical="4.0"/>
</Component>
<Component>
  <Name>ChrominanceRedDifference</Name>
  <Offset horizontal="0.0" vertical="2.5"/>
  <Period horizontal="4.0" vertical="2.0"/>
</Component>
</Field>
</ColorSampling>
</VisualCoding>
</MediaFormat>
</MediaProfile>
</MediaInformation>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

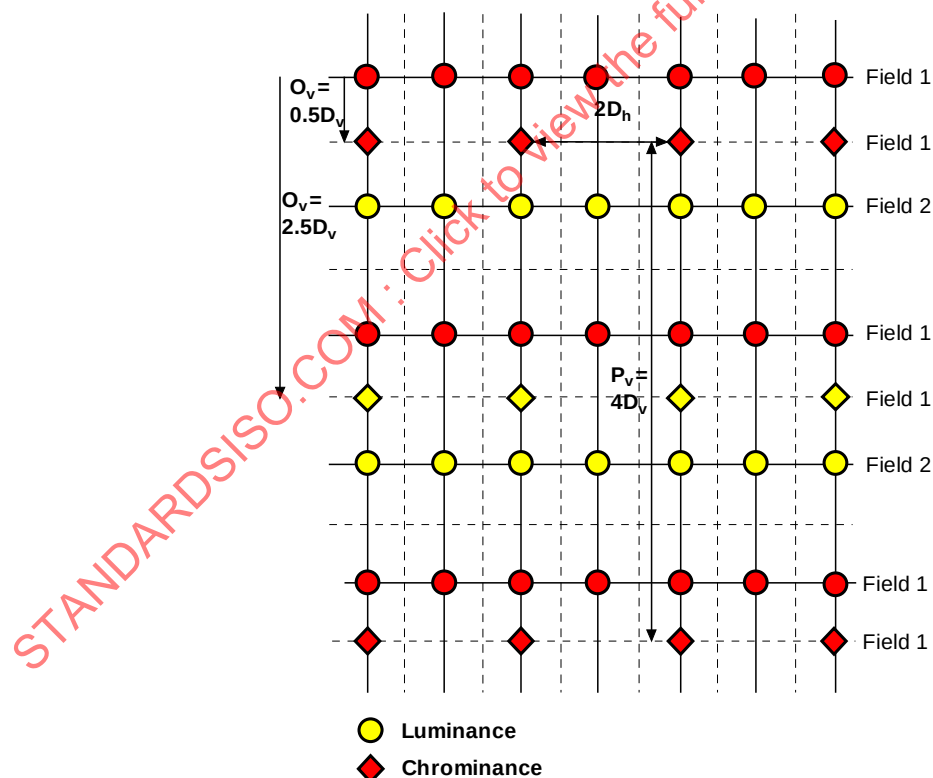


Figure 21 — Illustration of the siting of luminance and chrominance samples for 4:2:0 data (i.e., MPEG-2 interlaced color sampling YCbCr 4:2:0)

9 Creation and production description tools

9.1 Introduction

This clause specifies tools for describing information about the creation and production of multimedia content. The creation information is related to the multimedia content and is typically not depicted in the multimedia content. As a result, the creation information is typically not extracted from the multimedia content itself. The following tools are specified in this clause:

Table 13 — Overview of Creation and Production Tools

Tool	Function
Creation and Production Tools	Describes information about the creation and production of the multimedia content. The CreationInformation DS is composed of one Creation DS, zero or one Classification DS, and zero or more RelatedMaterial DSS. The Creation DS contains description tools for author generated information about the creation process. The Classification DS contains the description tools for classifying the multimedia content using classification schemes and also subjective reviews. The RelatedMaterial DS contains the description tools that allow the description of additional material that is related to the multimedia content.

9.2 Creation information tools

9.2.1 CreationInformation DS

9.2.1.1 Introduction

The CreationInformation DS, which describes information about the creation and production of multimedia content, contains the following information:

- **Creation and production information:** information about the creation and production of the content not perceived in the content, such as author, director, characters, target audience, and so forth, and information about the creation process that may be perceived in the content, such as actors in a video, players in a concert, and so forth.
- **Classification information:** information related to the classification of the content, such as target audience, style, genre, rating, and so forth.

9.2.1.2 CreationInformation DS syntax

```
<!-- ##### -->
<!-- Definition of CreationInformation DS (9.2.1) -->
<!-- ##### -->

<!-- Definition of CreationInformation DS -->
<complexType name="CreationInformationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Creation" type="mpeg7:CreationType"/>
        <element name="Classification" type="mpeg7:ClassificationType"
          minOccurs="0"/>
        <element name="RelatedMaterial" type="mpeg7:RelatedMaterialType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

    </extension>
  </complexContent>
</complexType>

```

9.2.1.3 CreationInformation DS semantics

Semantics of the CreationInformationType:

Name	Definition
CreationInformationType	Describes creation features of the multimedia content.
Creation	Describes by whom, when, and where, etc, the multimedia content was created.
Classification	Describes user oriented and service oriented classification of the multimedia content (optional).
RelatedMaterial	Describes additional information about material related to the multimedia content (optional).

9.2.2 Creation DS

9.2.2.1 Introduction

The Creation DS describes the creation of the multimedia content, including places, dates, actions, materials, staff (technical and artistic) and organizations involved. The creators defined in this subclause are for the purposes of description and discovery only, not for the purposes of Digital Rights Management (DRM).

9.2.2.2 Creation DS syntax

```

<!-- ##### -->
<!-- Definition of Creation DS (9.2.2) -->
<!-- ##### -->

<!-- Definition of Creation DS -->
<complexType name="CreationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Title" type="mpeg7:TitleType"
          minOccurs="1" maxOccurs="unbounded"/>
        <element name="TitleMedia" type="mpeg7:TitleMediaType"
          minOccurs="0"/>
        <element name="Abstract" type="mpeg7:TextAnnotationType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="Creator" type="mpeg7:CreatorType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="CreationCoordinates"
          minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="Location" type="mpeg7:PlaceType"
                minOccurs="0"/>
              <element name="Date" type="mpeg7:TimeType"
                minOccurs="0"/>

```

```

        </sequence>
      </complexType>
    </element>
    <element name="CreationTool" type="mpeg7:CreationToolType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="CopyrightString" type="mpeg7:TextualType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</extension>
</complexContent>
</complexType>

<!-- Definition of Title datatype -->
<complexType name="TitleType">
  <simpleContent>
    <extension base="mpeg7:TextualBaseType">
      <attribute name="type" use="optional" default="main">
        <simpleType>
          <union>
            <simpleType>
              <restriction base="NMTOKEN">
                <enumeration value="main"/>
                <enumeration value="secondary"/>
                <enumeration value="alternative"/>
                <enumeration value="original"/>
                <enumeration value="popular"/>
                <enumeration value="opusNumber"/>
                <enumeration value="songTitle"/>
                <enumeration value="albumTitle"/>
                <enumeration value="seriesTitle"/>
                <enumeration value="episodeTitle"/>
              </restriction>
            </simpleType>
            <simpleType>
              <restriction base="mpeg7:termReferenceType"/>
            </simpleType>
          </union>
        </simpleType>
      </attribute>
    </extension>
  </simpleContent>
</complexType>

<!-- Definition of TitleMedia datatype -->
<complexType name="TitleMediaType">
  <sequence>
    <element name="TitleImage" type="mpeg7:ImageLocatorType"
      minOccurs="0"/>
    <element name="TitleVideo" type="mpeg7:TemporalSegmentLocatorType"
      minOccurs="0"/>
    <element name="TitleAudio" type="mpeg7:TemporalSegmentLocatorType"
      minOccurs="0"/>
  </sequence>
</complexType>

<!-- Definition of Creator datatype -->
<complexType name="CreatorType">
  <complexContent>
    <extension base="mpeg7:MediaAgentType">
      <sequence>

```

```

        <element name="Character" type="mpeg7:PersonNameType"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="Instrument" type="mpeg7:CreationToolType"
            minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</extension>
</complexContent>
</complexType>

<!-- Definition of MediaAgent datatype -->
<complexType name="MediaAgentType">
    <sequence>
        <element name="Role" type="mpeg7:ControlledTermUseType"/>
        <choice>
            <element name="Agent" type="mpeg7:AgentType"/>
            <element name="AgentRef" type="mpeg7:ReferenceType"/>
        </choice>
    </sequence>
</complexType>

<!-- Definition of CreationTool datatype -->
<complexType name="CreationToolType">
    <sequence>
        <element name="Tool" type="mpeg7:TermUseType"/>
        <element name="Setting" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <attribute name="name" type="string" use="required"/>
                <attribute name="value" type="string" use="required"/>
            </complexType>
        </element>
    </sequence>
</complexType>

```

9.2.2.3 Creation DS semantics

Semantics of the CreationType:

Name	Definition
CreationType	Describes the creation of the content, including places, dates, actions, materials, staff (technical and artistic) and organizations involved.
Title	Describes one textual title of the multimedia content. Multiple titles are allowed. They may correspond to different types (indicated by the type attribute) or to different languages (indicated by the xml:lang attribute).
Titlemedia	Describes one multimedia title of the multimedia content (optional). It serves as a multimedia identifier of the multimedia content. It may contain a title in the form of an image, an audio clip, or a video.
Abstract	Describes a textual abstract of the multimedia content (optional). It is a summary, assigned during the creation process, of what is conveyed in the multimedia content.
Creator	Describes one creator of the multimedia content (optional). It allows the description of persons, organizations, groups, and so forth involved in the creation as well as their role.
CreationCoordinates	Describes the location and the date of creation of the multimedia content (optional).

Name	Definition
Location	Describes the place where the multimedia content was created (optional).
Date	Describes the date or period when the multimedia content was created (optional).
CreationTool	Describes one device (and its settings) used in the creation of the multimedia content (optional).
CopyrightString	Describes one textual label indicating information that may be displayed or otherwise made known to the end user (optional). It is not a formal declaration of the usage rights of the multimedia content.

Semantics of the TitleType:

Name	Definition
TitleType	Describes the textual title of multimedia content.
type	Indicates the type of the textual title of multimedia content. The types of titles are defined as follows. • <i>main</i> – The title is the main title given by the creator. This is the default value. • <i>secondary</i> – The title is a secondary title given by the creator. • <i>alternative</i> – The title is an alternative title given by the creator. • <i>original</i> – The title is the original title given by the creator. • <i>popular</i> – The title is a popular title given by the public and media. • <i>opusNumber</i> – The title correspond to the opus number. • <i>songTitle</i> – The title corresponds to a song. • <i>albumTitle</i> – The title corresponds to an album. • <i>seriesTitle</i> – The title corresponds to a series. • <i>episodeTitle</i> – The title corresponds to an episode of a series. It complements the series title.

Other values that are datatype-valid with respect to `mpeg7:termReferenceType` are reserved.

Semantics of the TitleMediaType:

Name	Definition
TitleMediaType	Specifies the title of the multimedia content using a multimedia document
TitleImage	Specifies the title of the multimedia content in image form (e.g, a thumbnail) (optional).
TitleVideo	Specifies the title of the multimedia content in video form (e.g, a video clip) (optional).

<i>Name</i>	<i>Definition</i>
TitleAudio	Specifies the title of the multimedia content in audio form (e.g, a jingle) (optional).

Semantics of the CreatorType:

<i>Name</i>	<i>Definition</i>
CreatorType	Describes the creator of the multimedia content or the description. The creator can be person(s), organization(s), or group(s) involved in the creation, with an associated role.
Character	Describes the name of a fictional character that is part of the multimedia content or that specifies a role played by an actor (optional). Multiple character descriptors may be associated with a creator.
Instrument	Describes the device, instrument, or tool used by the creator (optional).

Semantics of the MediaAgentType:

<i>Name</i>	<i>Definition</i>
MediaAgentType	Describes an agent with a role.
Role	Describes the role played by the media agent. An example of a ClassificationScheme is the RoleCS.
Agent	Describes the Agent (person, organization or group).
AgentRef	References the Agent (person, organization or group).

Semantics of the CreationToolType:

<i>Name</i>	<i>Definition</i>
CreationToolType	Describes the tool and settings used for the creation of the multimedia content or the description.
Tool	Describes a tool used during the creation process. For example: software, lens, films, musical instrument, microphone, and so forth.
Setting	Indicates the setting (parameters) for the tool used during creation (optional). The names and values of tool setting are tool specific and are not specified by this standard.
name	Indicates the name of the setting (parameter) for the tool.
value	Indicates the value of the specified parameter. All parameters are represented as string values. For example, the value may give the focal length of a camera or an input parameter for a software a tool.

9.2.3 Classification DS

9.2.3.1 Introduction

The Classification DS describes the classification of the multimedia content. The resulting descriptions facilitate searching and filtering of multimedia content based on user preferences (e.g, language, style, genre, and so forth) and service-oriented classifications (e.g, purpose, parental guidance, market segmentation, media review, and so forth).

9.2.3.2 Classification DS syntax

```

<!-- ##### -->
<!-- Definition of Classification DS (9.2.3) -->
<!-- ##### -->

<!-- Definition of Classification DS -->
<complexType name="ClassificationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Form" type="mpeg7:ControlledTermUseType" minOccurs="0"/>
        <element name="Genre" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:ControlledTermUseType">
                <attribute name="type" use="optional" default="main">
                  <simpleType>
                    <restriction base="NMTOKEN">
                      <enumeration value="main"/>
                      <enumeration value="secondary"/>
                    </restriction>
                  </simpleType>
                </attribute>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="Subject" type="mpeg7:TextAnnotationType"
          minOccurs="0"/>
        <element name="Purpose" type="mpeg7:ControlledTermUseType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="Language" type="mpeg7:ExtendedLanguageType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="CaptionLanguage" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="language">
                <attribute name="closed" type="boolean" use="optional"
                  default="true"/>
                <attribute name="supplemental" type="boolean"
                  use="optional" default="false"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
        <element name="SignLanguage" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="language">
                <attribute name="primary" type="boolean" use="optional"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        <attribute name="translation" type="boolean"
            use="optional"/>
    </extension>
</simpleContent>
</complexType>
</element>
<element name="Release" minOccurs="0">
    <complexType>
        <sequence>
            <element name="Region" type="mpeg7:regionCode"
                minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="date" type="mpeg7:timePointType"
            use="optional"/>
    </complexType>
</element>
<element name="Target" minOccurs="0">
    <complexType>
        <sequence>
            <element name="Market" type="mpeg7:ControlledTermUseType"
                minOccurs="0" maxOccurs="unbounded"/>
            <element name="Age" minOccurs="0">
                <complexType>
                    <attribute name="min" type="nonNegativeInteger"
                        use="optional"/>
                    <attribute name="max" type="nonNegativeInteger"
                        use="optional"/>
                </complexType>
            </element>
            <element name="Region" type="mpeg7:regionCode"
                minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
</element>
<element name="ParentalGuidance" type="mpeg7:ParentalGuidanceType"
    minOccurs="0" maxOccurs="unbounded"/>
<element name="MediaReview" type="mpeg7:MediaReviewType"
    minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</extension>
</complexContent>
</complexType>

<!-- Definition of ExtendedLanguage datatype -->
<complexType name="ExtendedLanguageType">
    <simpleContent>
        <extension base="language">
            <attribute name="type" use="optional" default="original">
                <simpleType>
                    <restriction base="NMTOKEN">
                        <enumeration value="original"/>
                        <enumeration value="dubbed"/>
                        <enumeration value="background"/>
                    </restriction>
                </simpleType>
            </attribute>
            <attribute name="supplemental" type="boolean"
                use="optional" default="false"/>
        </extension>
    </simpleContent>

```

```

</complexType>

<!-- Definition of ParentalGuidance datatype -->
<complexType name="ParentalGuidanceType">
  <sequence>
    <choice>
      <element name="ParentalRating" type="mpeg7:ControlledTermUseType"/>
      <element name="MinimumAge" type="nonNegativeInteger"/>
    </choice>
    <element name="Region" type="mpeg7:regionCode"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!-- Definition of MediaReview datatype -->
<complexType name="MediaReviewType">
  <sequence>
    <element name="Rating" type="mpeg7:RatingType" minOccurs="0"/>
    <element name="FreeTextReview" type="mpeg7:TextualType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="ReviewReference" type="mpeg7:RelatedMaterialType"
      minOccurs="0"/>
    <element name="Reviewer" type="mpeg7:AgentType" minOccurs="0"/>
  </sequence>
</complexType>

```

9.2.3.3 Classification DS semantics

Semantics of the ClassificationType:

Name	Definition
ClassificationType	Describes the classification of the multimedia content.
Form	Describes the production type of the document, such as, film, news program, magazine, documentary, etc. (optional). An example of CS is FormatCS.
Genre	Describes what the multimedia content is about (broad classification), such as sports, politics, economics, etc. (optional). An example of CS is the GenreCS.
type	Indicates the type of the genre of the multimedia content. The types of genres are defined as follows. <i>main</i> – The specified genre is the main, or primary. This is the default value. <i>secondary</i> – The specified genre is a secondary genre, such as a subgenre.
Subject	Describes the subject (specific classification) of the multimedia content (optional). The subject allows a textual annotation to classify the multimedia content.
Purpose	Describes one purpose for which the multimedia content was created (optional). An example of CS is IntentionCS.
Language	Describes one language of the spoken audio of the program (optional).
CaptionLanguage	Describes one language of the caption information included with the program (optional). The type of the caption information associated with the program is denoted by the closed attribute. Closed captions can be turned on or off by the user, while open captions (or subtitles) are part of the picture itself and remain visible.

Name	Definition
closed	Indicates whether the specified caption is closed. Default value of the attribute is <i>true</i> ; if the attribute is set to <i>false</i> , then the provided caption description refers to open captions/subtitles.
supplemental	Indicates whether the captions provide descriptions of the scene for the benefit of hearing or visually impaired, in addition to a direct translation of the spoken words. Closed captions may include such descriptive information, such as speaker identification, and non-speech sounds that would be missed. Default value of this attribute is <i>false</i> .
SignLanguage	Specifies the audio sign language provided for the multimedia content, and, optionally, qualifies the use of signing as a primary language or as a translation of the spoken dialogue (optional).
primary	Indicates if the sign language is the primary language of the content or not, i.e. if the content is produced specifically for the hearing impaired or not (optional).
translation	Indicates if the sign language is a translation of the spoken dialogue or not (optional).
Release	Describes the release date and region of the multimedia content (optional).
Region	Indicates the countries or regions in which the multimedia content was first released (optional). This locator may be different than the location where it was created.
date	Indicates the date on which the multimedia content was first released. This date may be different than the date(s) when it was created (optional).
Target	Describes the target of the multimedia content in terms of market classification, age and country or region (optional).
Market	Describes one targeted market of the multimedia content (optional). An example of CS is TargetGroupCS.
Age	Describes the targeted age range of the multimedia content (optional).
min	Indicated the minimum age of the targeted age range (optional).
max	Indicated the maximum age of the targeted age range (optional).
Region	Describes one target country or region for the multimedia content (optional).
ParentalGuidance	Describes one parental guidance classification of the multimedia content (optional).
MediaReview	Describes one media review about the multimedia content (optional).

Semantics of the ExtendedLanguageType:

Name	Definition
ExtendedLanguageType	Specifies the language of spoken audio of the multimedia content.
type	Indicates whether the spoken language is the original language of the content, or has been dubbed. The following values are allowed. <i>original</i> – The specified language is an original language of the spoken content. This is the default value. <i>dubbed</i> – The specified language is a dubbed language of the spoken content. <i>background</i> – The specified language is the original language heard in the background, while another (dubbed) language is heard in the foreground. Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.
supplemental	Indicates whether the specified language exclusively provides an audio description of the multimedia content, for the benefit of the visually impaired. Default value of this attribute is <i>false</i> .

Semantics of the ParentalGuidanceType:

Name	Definition
ParentalGuidanceType	Describes parental guidance classification.
ParentalRating	Indicates the rating value according to the rating scheme. Examples of CSs are JapanRatingCS, ICRAParentalRatingCS, MPAAParentalRatingCS, ICRAParentalRatingNudityCS, RIAAParentalRatingCS, ICRAParentalRatingSexCS, MPAAParentalRatingTVCS, and ICRAParentalRatingViolenceCS.
MinimumAge	Indicates the minimum recommended age in years of the end user.
Region	Indicates the country or region for which the parental guidance description is specified (optional). If not specified, the parental guidance applies worldwide.

Semantics of the MediaReviewType:

Name	Definition
MediaReviewType	Describes the Media Review about the multimedia content.
Rating	Specifies the rating value and criterion used in the review (optional).
FreeTextReview	Describes a free-text review of the multimedia content without reference to a rating scheme (optional). There can be multiple instances of the review in different languages.

Name	Definition
ReviewReference	Describes the Media Locator of the material from where the review may have been extracted or quoted (optional). For example, the TV magazine that published the review, the Films Guide from where the review was quoted, an interview from where the review was transcribed.
Reviewer	Describes the reviewer/critic of the multimedia content (optional). Since it is defined as AgentType, this field can describe a person, a group of persons or an organization.

9.2.4 RelatedMaterial DS

9.2.4.1 Introduction

The RelatedMaterial DS describes additional information about the multimedia content available in other materials.

9.2.4.2 RelatedMaterial DS syntax

```

<!-- ##### -->
<!-- Definition of RelatedMaterial DS (9.2.4) -->
<!-- ##### -->

<!-- Definition of RelatedMaterial DS -->
<complexType name="RelatedMaterialType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="DisseminationFormat"
          type="mpeg7:ControlledTermUseType" minOccurs="0"/>
        <element name="MaterialType" type="mpeg7:TermUseType"
          minOccurs="0"/>
        <choice>
          <element name="MediaLocator" type="mpeg7:MediaLocatorType"/>
          <element name="MediaInformation" type="mpeg7:MediaInformationType"/>
          <element name="MediaInformationRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0">
          <element name="CreationInformation"
            type="mpeg7:CreationInformationType"/>
          <element name="CreationInformationRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0">
          <element name="UsageInformation" type="mpeg7:UsageInformationType"/>
          <element name="UsageInformationRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

9.2.4.3 RelatedMaterial DS semantics

Semantics of the RelatedMaterialType:

Name	Definition
RelatedMaterialType	Describes material containing additional information about the multimedia content or related to it (e.g., extended reports of a news program or Web pages with information about topics covered in the news).
DisseminationFormat	Describes the publication medium or delivery mechanism of the related material, for example, terrestrial broadcast, web-cast, streaming, CD-ROM, and so forth. An example of CS is DisseminationFormatCS.
MaterialType	Describes the type of the related material (optional). For example, script of a movie, original book, lyrics of a song, or drafts of a graphic design.
MediaLocator	Describes the media location of the related material.
MediaInformation	Describes the media information about the related material. This descriptor provides information sufficient for locating the media.
MediaInformationRef	References the media information description of the related material. This descriptor provides information sufficient for locating the media.
CreationInformation	Describes the creation information about the related material.
CreationInformationRef	References the creation information description of the related material.
UsageInformation	Describes the usage information about the related material.
UsageInformationRef	References the usage information description of the related material.

9.2.5 CreationInformation examples (informative)

9.2.5.1 CreationInformation DS examples

The following example shows the use of CreationInformation DS for describing the creation information for a video.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <CreationInformation id="news1_creation">
          <Creation>
            <Title xml:lang="es">Telediario (segunda edicion)</Title>
            <Title xml:lang="en" type="popular">Afternoon news</Title>
            <Title xml:lang="es" type="popular">Noticias de la tarde</Title>
            <Abstract>
              <FreeTextAnnotation>
                The trial about the kidnapping of Segundo Marey started this
                morning. Short interview with Segundo Marey after the end of the
                first session.
              </FreeTextAnnotation>
              <StructuredAnnotation>
                <Who>
                  <Name>Segundo Marey</Name>
                </Who>
              </StructuredAnnotation>
            </Abstract>
          </Creation>
        </CreationInformation>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```



```

        </Who>
    </StructuredAnnotation>
</Abstract>
<Creator>
    <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:ANCHOR">
        <Name xml:lang="en">Anchorman</Name>
    </Role>
    <Agent xsi:type="PersonType">
        <Name>
            <GivenName>Ana</GivenName>
            <FamilyName>Blanco</FamilyName>
        </Name>
    </Agent>
</Creator>
<CreationCoordinates>
    <Location>
        <Name xml:lang="es">Piruli</Name>
        <Region>es</Region>
        <AdministrativeUnit>Madrid</AdministrativeUnit>
    </Location>
    <Date>
        <TimePoint>1998-10-03T14:13+01:00</TimePoint>
    </Date>
</CreationCoordinates>
<CreationTool>
    <Tool>
        <Name>Video Camera</Name>
    </Tool>
    <Setting name="zoom" value="1.5"/>
    <Setting name="aperture" value="35"/>
</CreationTool>
</Creation>
<Classification>
    <Form href="urn:mpeg:mpeg7:cs:FormatCS:2001:1.1">
        <Name xml:lang="en">Bulletin</Name>
    </Form>
    <Genre href="urn:mpeg:mpeg7:cs:ContentCS:2001:1.3.1">
        <Name xml:lang="en">Daily news</Name>
    </Genre>
    <Purpose href="urn:mpeg:mpeg7:cs:IntentionCS:2001:2.1">
        <Name xml:lang="en">Pure information</Name>
    </Purpose>
    <Language type="original">es</Language>
    <Release date="1998-06-16T21:30+01:00">
        <Region>es</Region>
    </Release>
    <ParentalGuidance>
        <ParentalRating
            href="urn:mpeg:mpeg7:cs:MPAAParentalRatingCS:2001:G"/>
        <Region>es</Region>
    </ParentalGuidance>
    <MediaReview>
        <Rating>
            <RatingValue>10</RatingValue>
            <RatingScheme best="10" worst="1" style="higherBetter">
                <Name>Overall</Name>
            </RatingScheme>
        </Rating>
    </MediaReview>
    <MediaReview>
        <FreeTextReview xml:lang="en">Interesting News today.

```

```

        </FreeTextReview>
        <FreeTextReview xml:lang="es">Interesantes noticias hoy.
        </FreeTextReview>
        <ReviewReference>
            <DisseminationFormat
                href="urn:mpeg:mpeg7:cs:DisseminationFormatCS:2001:4">
                <Name xml:lang="en">Internet</Name>
            </DisseminationFormat>
            <MediaLocator>
                <MediaUri>http://tvmagazine.com</MediaUri>
            </MediaLocator>
        </ReviewReference>
    </MediaReview>
</Classification>
<RelatedMaterial>
    <DisseminationFormat
        href="urn:mpeg:mpeg7:cs:DisseminationFormatCS:2001:4"/>
    <MaterialType>
        <Name xml:lang="en">Broadcaster Web Page</Name>
    </MaterialType>
    <MediaLocator>
        <MediaUri>http://www.rtve.es</MediaUri>
    </MediaLocator>
</RelatedMaterial>
</CreationInformation>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

9.2.5.2 Additional language description examples

In the following example, the original spoken audio of the multimedia content is in English; however, the content includes dubbed audio tracks in French and Spanish.

```

<CreationInformation>
<!-- more elements here -->
    <Classification>
        <Language type="original">en</Language>
        <Language type="dubbed">es</Language>
        <Language type="dubbed">fr</Language>
    </Classification>
</CreationInformation>

```

The following example is a description of a program broadcasted by a French TV station. The interview was originally done in Spanish and dubbed to English for a program produced and broadcasted in England. After dubbing, the Spanish can still be heard in the background. The interview was bought by the French TV station, which added subtitles in French for its broadcast.

```

<CreationInformation>
<!-- more elements here -->
    <Classification>
        <Language type="dubbed">en</Language>
        <Language type="background">es</Language>
        <CaptionLanguage closed="false">fr</CaptionLanguage>
    </Classification>
</CreationInformation>

```

10 Usage description tools

10.1 Introduction

This clause specifies tools describing information about the usage of the multimedia content. The following tools are specified in this clause:

Table 14 — Overview of Usage Information Tools

Tool	Functionality
Usage Tools	Describes information about the usage of the multimedia content. The UsageInformation DS includes a Rights D, zero or one FinancialResults D, and several Availability DSs and associated UsageRecord DSs. The Rights D contains references to rights holders in order to obtain rights information. The FinancialResults D contains the description of costs and incomes generated by the multimedia content. The Availability DS describes the details about the availability of the multimedia content for usage. The UsageRecord DS describes the details pertaining to the usage of the multimedia content.

10.2 Usage information tools

10.2.1 UsageInformation DS

10.2.1.1 Introduction

The UsageInformation DS describes information about the usage of the multimedia content. The UsageInformation DS describes:

- **Rights information:** information about the rights for using the multimedia content.
- **Usage information:** information about the ways and means to use the multimedia content (e.g. edition, emission, and so forth) and the results of the usage (e.g. audience)
- **Financial information:** information about the financial results of the production (in the FinancialResults D within the UsageInformation DS), and of the publication (in the Financial D within each Availability DS and UsageRecord DS) of the multimedia content.

Note: new descriptions may be added each time the content is used (e.g. UsageRecord, Income), or when there are new ways to access to the multimedia content for some kind of applications, if required. In such case, the size of the description might significantly increase. This information should be taken into account when designing the systems and applications using these descriptions.

10.2.1.2 UsageInformation DS syntax

```
<!-- ##### -->
<!-- Definition of UsageInformation DS (10.2.1) -->
<!-- ##### -->

<!-- Definition of UsageInformation DS -->
<complexType name="UsageInformationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Rights" type="mpeg7:RightsType"
          minOccurs="0"/>
        <element name="FinancialResults" type="mpeg7:FinancialType"
```

```

        minOccurs="0"/>
        <element name="Availability" type="mpeg7:AvailabilityType"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="UsageRecord" type="mpeg7:UsageRecordType"
            minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</extension>
</complexContent>
</complexType>

```

10.2.1.3 UsageInformation DS semantics

Semantics of the UsageInformationType:

Name	Definition
UsageInformationType	Describes usage features of the multimedia content.
Rights	Describes information about the owners of the rights corresponding to the multimedia content, and how the multimedia content can be used (optional). Its appearance at this level precludes its appearance in the Availability DS instance of the same UsageInformation instance.
FinancialResults	Describes the cost of the creation of the multimedia content and the income the multimedia content has generated (optional). The incomes vary with time.
Availability	Describes where, when, how, and by whom the multimedia content can be used (optional).
UsageRecord	Describes where, when, how, and by whom the multimedia content was used (optional).

10.2.2 Rights datatype

10.2.2.1 Introduction

The Rights Datatype describes links to the rights holders of the multimedia content (IPR) and the Access Rights.

Ownership of rights is dynamic as it can pass from one rights holder to another. For example, the rights to a television production (or parts thereof) are often licensed and assigned. To enable systems to be able to discover the current rights situation, the Rights Datatype carries an identifier that can be resolved through an appropriate agency.

For the purpose of rights management, such agencies and systems are progressively being extended to enable rights holders to declare the identity of the current rights holder.

10.2.2.2 Rights datatype syntax

```

<!-- ##### -->
<!-- Definition of Rights datatype (10.2.2) -->
<!-- ##### -->

<!-- Definition of Rights datatype -->
<complexType name="RightsType">

```

```

<sequence>
  <element name="RightsID" type="mpeg7:UniqueIDType"
    minOccurs="1" maxOccurs="unbounded"/>
</sequence>
</complexType>

```

10.2.2.3 Rights datatype semantics

Semantics of the RightsType:

Name	Definition
RightsType	Describes a link to the right holders and to the access rights information. It gives an ID (such as specified in the MPEG-4 IP Identification Data Set) that enables access to the current Rights Owner information about the multimedia content or the description (making use of appropriate rights management databases).
RightsID	Identifies the link to the current Rights information about the content, including Rights Holders, Access Rights, and other related information.

10.2.3 Financial datatype

10.2.3.1 Introduction

The Financial Datatype contains information related to the costs generated and incomes produced by multimedia content. The notions of partial costs and incomes allow to classify the different costs and incomes by their type. As the type is defined as a controlled term, several standardized types of costs and incomes can be used. Total and subtotal costs and incomes are to be calculated by the application from these partial values.

10.2.3.2 Financial datatype syntax

```

<!-- ##### -->
<!-- Definition of Financial datatype (10.2.3) -->
<!-- ##### -->

<!-- Definition of Financial datatype -->
<complexType name="FinancialType">
  <sequence>
    <element name="AccountItem" minOccurs="1" maxOccurs="unbounded" >
      <complexType>
        <sequence>
          <element name="EffectiveDate"
            type="mpeg7:timePointType" minOccurs="0"/>
          <choice>
            <element name="CostType" type="mpeg7:TermUseType"/>
            <element name="IncomeType"
              type="mpeg7:TermUseType"/>
          </choice>
        </sequence>
        <attribute name="currency" type="mpeg7:currencyCode"
          use="required"/>
        <attribute name="value" type="decimal" use="required"/>
      </complexType>
    </element>
  </sequence>
</complexType>

```

10.2.3.3 Financial datatype semantics

Semantics of the FinancialType:

Name	Definition
FinancialType	Describes the costs and incomes generated by the multimedia content.
AccountItem	Describes an account item (cost or income).
EffectiveDate	Describes when the cost or income is made effective (optional). It can be the same as when it was generated or not.
CostType	Describes the type of a cost.
IncomeType	Describes the type of an income.
currency	Describes the currency of the price (see ISO 4217).
value	Describes the value of the price.

10.2.4 Availability DS

10.2.4.1 Introduction

The Availability DS describes the availability for use (broadcasting, on demand delivery, CD sales, and so forth) of the multimedia content. The Availability DSs are referenced by the UsageRecord DSs keeping track of the usage of the multimedia content.

10.2.4.2 Availability DS syntax

```

<!-- ##### -->
<!-- Definition of Availability DS (10.2.4) -->
<!-- ##### -->

<!-- Definition of Availability DS -->
<complexType name="AvailabilityType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="InstanceRef" type="mpeg7:ReferenceType"/>
        <element name="Dissemination" type="mpeg7:DisseminationType"
          minOccurs="0"/>
        <element name="Financial" type="mpeg7:FinancialType" minOccurs="0"/>
        <element name="Rights" type="mpeg7:RightsType" minOccurs="0"/>
        <element name="AvailabilityPeriod" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:TimeType">
                <attribute name="type" use="optional">
                  <simpleType>
                    <list>
                      <simpleType>
                        <union>
                          <simpleType>
                            <restriction base="NMTOKEN">
                              <enumeration value="live"/>

```

```

        <enumeration value="repeat"/>
        <enumeration value="firstShowing"/>
        <enumeration value="lastShowing"/>
        <enumeration
            value="conditionalAccess"/>
        <enumeration value="encrypted"/>
        <enumeration value="payPerUse"/>
    </restriction>
</simpleType>

    <simpleType>
        <restriction
            base="mpeg7:termReferenceType"/>
    </simpleType>
</union>
</simpleType>
</list>
</simpleType>
</attribute>
</extension>
</complexContent>
</complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

<!-- Definition of Dissemination DS -->
<complexType name="DisseminationType">
    <complexContent>
        <extension base="mpeg7:DSType">
            <sequence>
                <element name="Source" type="mpeg7:TextualType" minOccurs="0"/>
                <element name="Format" type="mpeg7:ControlledTermUseType"
                    minOccurs="0"/>
                <element name="Disseminator" type="mpeg7:MediaAgentType"
                    minOccurs="0"/>
                <element name="Location" type="mpeg7:PlaceType" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

10.2.4.3 Availability DS semantics

Semantics of the AvailabilityType:

Name	Definition
AvailabilityType	Describes the availability of the multimedia content. For example, broadcasting, on demand delivery, CD sales, and so forth.
InstanceRef	Describes a reference to the MediaInstance for which the availability is described.
Dissemination	Describes how the multimedia content is disseminated (optional).

Name	Definition
Financial	Describes the financial information related to the particular use described in the Availability description (optional). For example, it allows the description of the price of using the content and additional costs due to the availability of the content.
Rights	Describes information about the owners of the rights corresponding to the multimedia content, and how the multimedia content can be used (optional). Its appearance at this level precludes its appearance in the Rights instance of the same UsageInformation instance.
AvailabilityPeriod	Describes one period (date, time, and duration) of availability of the multimedia content (optional). For example, time and date of the broadcast, or time and date of availability for an on-demand content. In the case of availability on a persistent medium (e.g., DVD), it specifies the date of publication. If there is no duration description, the duration of the availability is unbounded.
type	Indicates the types of availability of the period (optional): • <i>live</i> – Indicates the multimedia content is made available (e.g., broadcast) live. • <i>repeat</i> – Indicates the multimedia content is a repeat of multimedia content that has been made available earlier (e.g., broadcast). • <i>firstShowing</i> – Indicates the given availability period is the first showing of the multimedia content. • <i>lastShowing</i> – Indicates the given availability period is the final showing of the multimedia content. • <i>conditionalAccess</i> – Indicates the access to the multimedia content is restricted by a conditional access mechanism. • <i>encrypted</i> – Indicates the multimedia content is encrypted for restricted viewing. • <i>payPerUse</i> – Indicates the multimedia content is pay-per-use (e.g., pay-per-view). Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.

Semantics of the DisseminationType:

Name	Definition
DisseminationType	Describes the dissemination mechanism of the multimedia content.
Source	Describes the source of the content, referring to, for example, a broadcast channel or a server.
Format	Describes the publication medium or delivery mechanism of the multimedia content, for example, terrestrial broadcast, web-cast, streaming, CD-ROM, etc (optional). An example of CS is DisseminationFormatCS.
Disseminator	Describes the disseminator of the multimedia content (optional). The disseminator refers to the agent or organization that makes the content available.

Name	Definition
Location	Describes the place from which the multimedia content is disseminated (optional).

10.2.5 UsageRecord DS

10.2.5.1 Introduction

The UsageRecord DS describe the past use of the multimedia content.

10.2.5.2 UsageRecord DS syntax

```

<!-- ##### -->
<!-- Definition of UsageRecord DS (10.2.5) -->
<!-- ##### -->

<!-- Definition of UsageRecord DS -->
<complexType name="UsageRecordType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="AvailabilityRef" type="mpeg7:ReferenceType"/>
        <element name="Audience" type="nonNegativeInteger" minOccurs="0"/>
        <element name="Financial" type="mpeg7:FinancialType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

10.2.5.3 UsageRecord DS semantics

Semantics of the UsageRecordType:

Name	Definition
UsageRecordType	Describes past use of the multimedia content.
AvailabilityRef	Describes a reference to the content availability's description.
Audience	Specifies the number of users who used the multimedia content in the particular use described in the referenced Availability description (optional).
Financial	Describes the financial information related to the particular use that is described in the referenced availability description (optional).

10.2.6 Usage information examples (informative)

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <UsageInformation id="news1_usage">
          <Rights>
            <RightsID organization="TVE" type="TVECopyRightsID">
              tve:19980618:td2
            </RightsID>
          </Rights>
          <FinancialResults>
            <AccountItem currency="EUR" value="423.46">
              <CostType>
                <Name>Total for production</Name>
              </CostType>
            </AccountItem>
            <AccountItem currency="EUR" value="200">
              <CostType>
                <Name>Broadcast</Name>
              </CostType>
            </AccountItem>
            <AccountItem currency="EUR" value="600">
              <IncomeType>
                <Name>Advertisement</Name>
              </IncomeType>
            </AccountItem>
          </FinancialResults>
          <Availability id="live_news1">
            <InstanceRef href="#onlinemp7cs17news1"/>
            <Dissemination>
              <Format>
                href="urn:mpeg:mpeg7:cs:DisseminationFormatCS:2001:4">
                <Name xml:lang="en">Internet</Name>
              </Format>
              <Location>
                <Region>es</Region>
              </Location>
            </Dissemination>
          </Availability>
          <Financial>
            <AccountItem currency="EUR" value="1">
              <CostType>
                <Name>Pay-per-view price</Name>
              </CostType>
            </AccountItem>
          </Financial>
          <AvailabilityPeriod type="payPerUse">
            <TimePoint>1998-06-16T21:00+01:00</TimePoint>
            <Duration>PT30M</Duration>
          </AvailabilityPeriod>
        </Availability>
        <Availability id="repeated_news1">
          <InstanceRef href="http://www.rtve.es/19980616/td2.mpeg"/>
          <Dissemination>
            <Format>
              href="urn:mpeg:mpeg7:cs:DisseminationFormatCS:2001:1">
              <Name>Terrestrial</Name>
            </Format>
            <Disseminator>
              <Role href="urn:mpeg:mpeg7:cs:RoleCS:2001:PUBLISHER"/>
            </Disseminator>
          </Dissemination>
        </Availability>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

```

        <Agent xsi:type="OrganizationType">
          <Name>Discovery Channel</Name>
        </Agent>
      </Disseminator>
      <Location>
        <Region>es</Region>
      </Location>
    </Dissemination>
    <AvailabilityPeriod type="repeat">
      <TimePoint>1998-06-16T21:30+01:00</TimePoint>
    </AvailabilityPeriod>
  </Availability>
  <UsageRecord>
    <AvailabilityRef idref="live_news1"/>
    <Audience>245747</Audience>
    <Financial>
      <AccountItem currency="EUR" value="1">
        <EffectiveDate>1998-06-16</EffectiveDate>
        <IncomeType>
          <Name>Income for pay-per-view live access</Name>
        </IncomeType>
      </AccountItem>
    </Financial>
  </UsageRecord>
</UsageInformation>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

11 Structure description tools

11.1 Introduction

This clause specifies tools for describing the structure of multimedia content. These tools describe the following:

- Images and spatial regions of images and video frames;
- Video, temporal segments, and spatio-temporal segments of video;
- Audio and temporal segments of audio;
- Audio-visual content and spatio-temporal segments of AV content;
- Multimedia content and segments of multimedia content;
- Visual text in images or video;
- Video mosaics;
- Edited video and segments of edited video;
- Ink content.

These tools apply to both digital and analog forms of multimedia content. These tools describe the following properties and relationships of the segments:

- Segment attributes related to creation and production (see clause 9), media information (see clause 8), usage information (see clause 10), semantic information (see clause 12), spatial and temporal localization, and spatial and temporal masks;
- Structural decompositions of the segments;
- Structural relations including relations among the segments.

11.1.1 Overview

The following tools are specified in this clause:

Table 15 — Overview of the structure description tools

<i>Tool</i>	<i>Functionality</i>
Base segment description tools	Abstract base for describing the structure of multimedia content.
Segment attribute description tools	Describes attributes of segments.
Visual segment description tools	Describes segments of visual content such as images and video.
Audio segment description tools	Describes segments of audio content.
Audio-visual segment description tools	Describes segments of AV content.
Multimedia segment description tools	Describes segments of multimedia content.
Ink segment description tools	Describes segments of ink content.
Video editing segment description tools	Describes segments of edited video.
Structural relation classification schemes	Describes spatial and temporal relations.

Figure 22 illustrates the organization of structure description tools. These description tools are based on the concept of a segment, which is a spatial and/or temporal unit of multimedia. For example, a video segment corresponds to a temporal unit or group of temporal units of video. While the structure description tools allow description of individual segments of multimedia, these tools should also be used to describe full multimedia content. That is, the tools should be used for describing a video as well as a video segment.

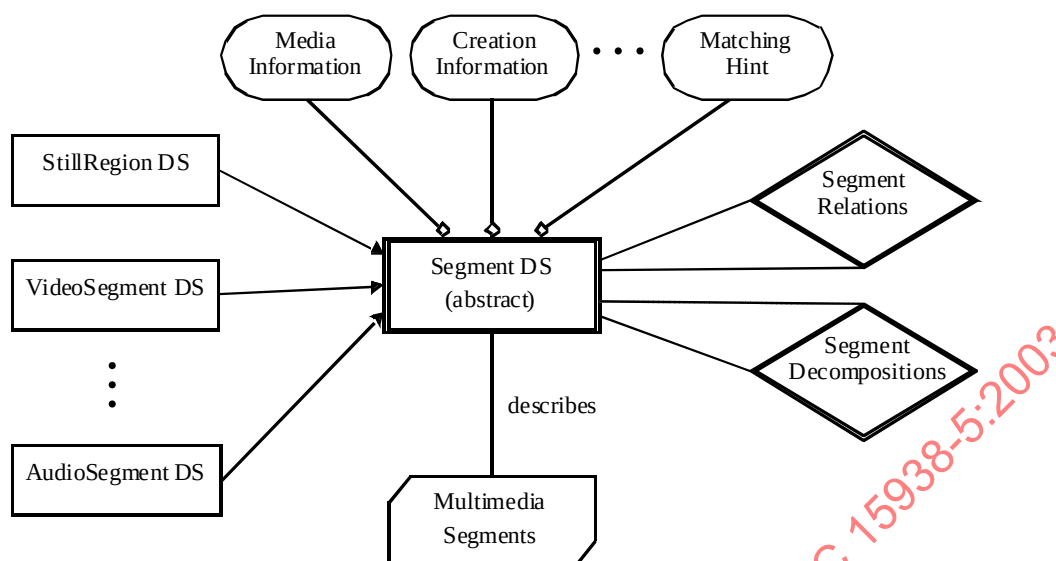


Figure 22 — Illustration of the organization of the tools for describing the structure of multimedia content

As shown in Figure 22, the Segment DS is the abstract tool that is the base of the specialized segment description tools. The specialized segment description tools extend the base Segment DS to describe the structure of specific types of multimedia content and their segments. Examples include still regions, video segments, audio segments, moving regions, and so forth. Figure 23 provides an overview of the specific tools for describing different types of segments.

The segment description tools are summarized as follows:

<i>Tool</i>	<i>Functionality</i>
Segment DS	Abstract base of the specialized segment description tools.
StillRegion DS	Describes a 2D spatial region of an image or video frame. A still region can be the entire image.
ImageText DS	Describes still regions of embedded or superimposed text in images or video frames.
Mosaic DS	Describes mosaics or panoramic views of video segments.
StillRegion3D DS	Describes 3D content and regions of 3D content.
VideoSegment DS	Describes video content and temporal segments of video content.
MovingRegion DS	Describes moving spatio-temporal regions of video content.
VideoText DS	Describes moving regions of embedded or superimposed text in video content.
AudioSegment DS	Describes audio content and temporal segments of audio content.
AudioVisualSegment DS	Describes AV content and temporal segments of AV content.
AudioVisualRegion DS	Describes spatio-temporal regions of AV content.

<i>Tool</i>	<i>Functionality</i>
MultimediaSegment DS	Describes multimedia content and the segments that form multimedia content.
InkSegment DS	Describes ink content and segments of ink content.
AnalyticEditedVideoSegment DS	Abstract base of the specialized segment DSs that describe edited video content from the analytic viewpoint.
EditedVideo DS	Describes edited video content and the segments of edited video content.
AnalyticClip DS	Abstract base of the specialized segment DSs that describe edited video clips from the analytic viewpoint.
Shot DS	Describes video segments that are delimited by global transitions.
CompositionShot DS	Describes video segments that are delimited by composition transitions.
IntraCompositionShot DS	Describes video segments that are delimited by internal transitions.
AnalyticTransition DS	Abstract base of the specialized segment DSs that describe transitions between video clips from the analytic viewpoint.
GlobalTransition DS	Describes global transitions of video content.
CompositionTransition DS	Describes composition transitions of video content.
InternalTransition DS	Describes internal transitions of video content.
EditedMovingRegion DS	Describes moving regions that correspond to edited video areas containing rushes or synthetic clips.

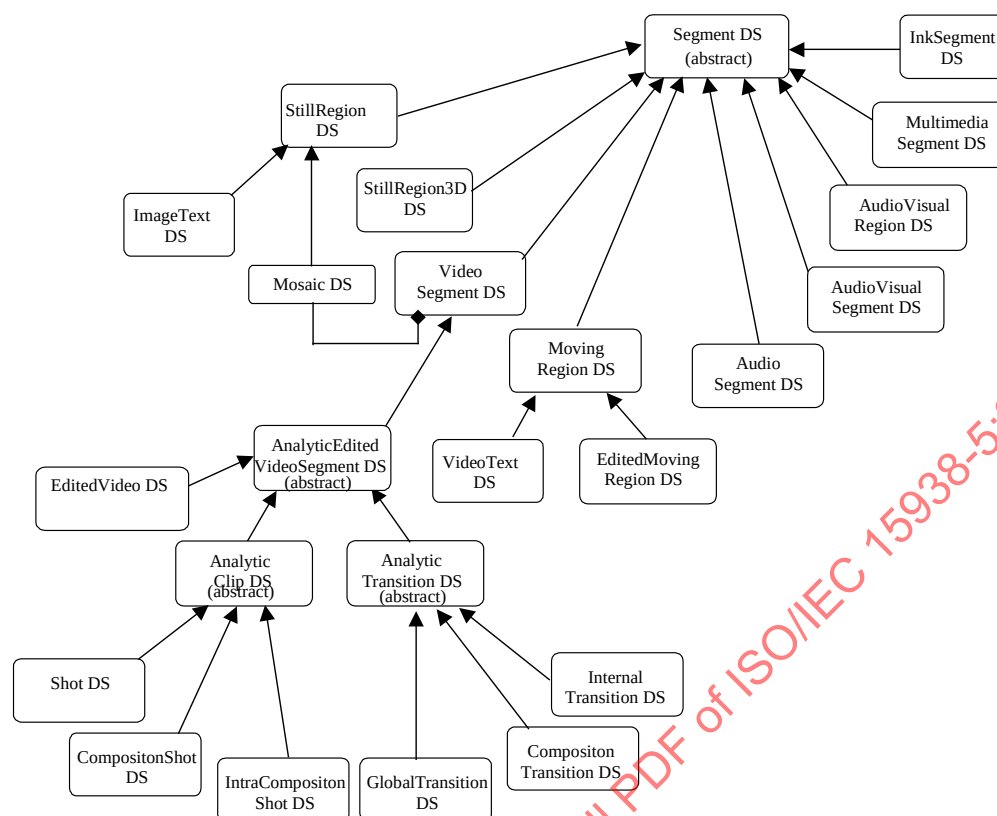


Figure 23 — Overview of the tools for describing segments of multimedia content

The segment attribute description tools describe the properties of segments such as creation information, media information, masks, matching hints, audio and visual features, and so forth.

The segment attribute description tools are summarized as follows:

Tool	Functionality
CreationInformation DS	Describes the creation and production information related to the segment. The CreationInformation DS is defined in clause 9.
MediaInformation DS	Describes the media information related to the segment. The MediaInformation DS is defined in clause 8.
UsageInformation DS	Describes the usage information related to the segment. The UsageInformation DS is defined in clause 10.
Semantic DS	Describes the semantic information related to the segment. The Semantic DS is defined in clause 12.
TextAnnotation datatype	Describes text that annotates or describes the content of the segment. The TextAnnotation datatype is defined in clause 7.3.3.
Mask D	Abstract base of the specialized mask Ds.
SpatialMask D	Describes a mask in 2-D space.
TemporalMask D	Describes a mask in time.

<i>Tool</i>	<i>Functionality</i>
SpatioTemporalMask D	Describes a mask in space and time.
MediaSpaceMask D	Describes a mask as a sequence of bytes.
SceneGraphMask D	Describes a mask of sub-graphs in a scene graph.
OrderedGroupDataSetMask D	Describes a mask from an ordered group dataset.
MatchingHint D	Describes hints for the matching of segments.
PointOfView D	Describes the relative importance of segments from a particular point of view.
InkMediaInformation DS	Describes media and creation information specific to ink content.
HandWritingRecogInformation DS	Describes hand-writing recognition information specific to ink content.
HandWritingRecogResult DS	Describes hand-writing recognition results information specific to ink content.

The segment decomposition tools describe the structural decomposition of segments of multimedia content. The SegmentDecomposition DS is the abstract tool that is the base of the specialized segment decomposition tools. The specialized decomposition tools extend the base SegmentDecomposition DS to describe the decomposition of specific types of multimedia content and their segments. Examples include spatial, temporal, spatio-temporal, and media source decompositions. Figure 24 provides an overview of the specific tools for describing different types of segment decompositions.

The segment decomposition tools are summarized as follows:

<i>Tool</i>	<i>Functionality</i>
Base segment decomposition tools	Abstract base of the specialized segment decomposition DSs.
Still region decomposition tools	Describes decompositions of still regions in space.
3D still region decomposition tools	Describes decompositions of 3D still regions in space.
Video segment decomposition tools	Describes decompositions of video segments in space, time, and media.
Moving region decomposition tools	Describes decompositions of moving regions in space, time, and media.
Ink segment decomposition tools	Describes decompositions of ink segments in space and time.
Audio segment decomposition tools	Describes decompositions of audio segments in time and media.
Audio-visual segment decomposition tools	Describes decompositions of AV segments in space, time, and media.
Audio-visual region decomposition tools	Describes decompositions of AV regions in space, time, and media.
Multimedia decomposition tools	Describes decompositions of multimedia segments in media.
Analytic edited video segment	Describes decompositions of analytic edited video segments in

<i>Tool</i>	<i>Functionality</i>
decomposition tools	space and time.
Edited video decomposition tools	Describes decompositions of edited video in time.
Shot decomposition tools	Describes decompositions of video shots in time.
Composition shot decomposition tools	Describes decompositions of video composition shots in time.
Intra-composition shot decomposition tools	Describes decompositions of video intra-composition shots in time.

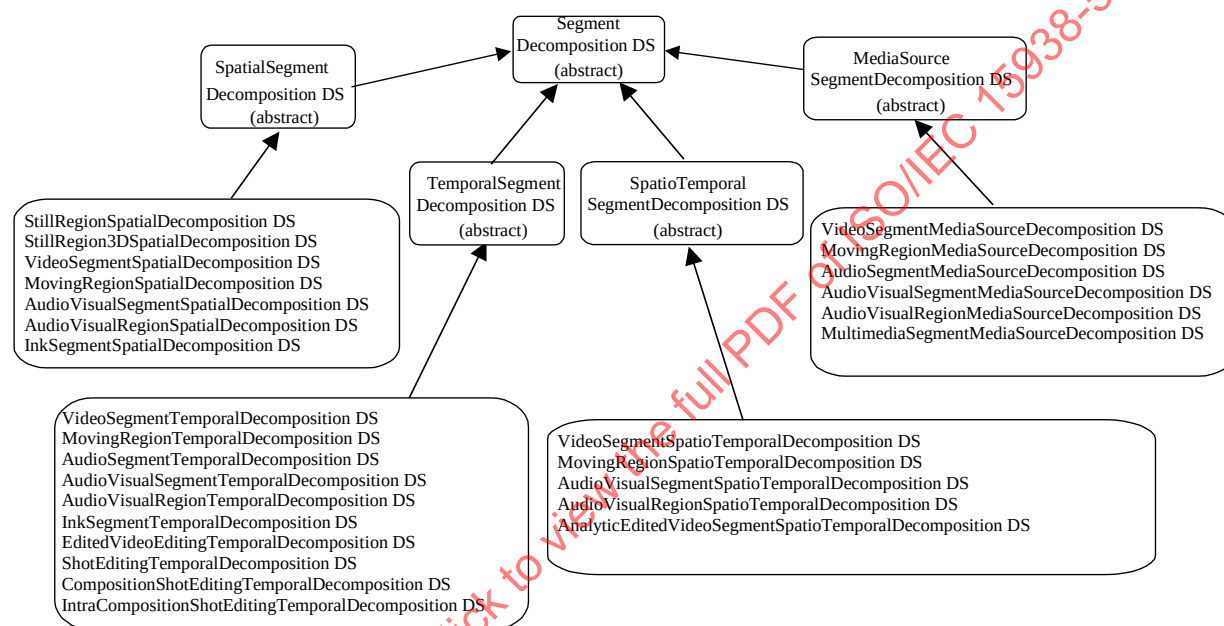


Figure 24 — Overview of the tools for describing structural decompositions of segments

The structural relation classification schemes describe spatial and temporal relations. The structural relations are defined in two classification schemes: the TemporalRelation CS and the SpatialRelation CS. The temporal and spatial relations apply to entities that have temporal or spatial attributes, respectively, which may be implicit or explicitly stated. The structural relation classification schemes should be used to describe the spatial and temporal relations among segments and semantic entities (see clause 12).

The structural relation classification schemes are summarized as follows:

<i>Classification Scheme</i>	<i>Functionality</i>
TemporalRelation CS	Describes temporal relations, which apply to entities that have time attributes, which may be implicit or explicitly stated.
SpatialRelation CS	Describes spatial relations, which apply to entities that have 2-D spatial attributes, which may be implicit or explicitly stated.

11.1.2 Structure description example

Figure 25 shows an example of the use of the structure description tools for describing a video. In this example, the video is described using the VideoSegment DS (top). The temporal decomposition of the video into two video segments is described using the VideoSegmentTemporalDecomposition DS. The specific attributes of the different video segments are described using the segment attribute description tools. The following tools are used to describe the attributes of the whole video (top): CreationInformation DS, MediaInformation DS, UsageInformation DS, and Semantic DS. The following tools are used to describe the attributes of the moving region on the left: TextAnnotation datatype, MediaTime datatype, MotionActivity D, and other visual Ds. The following tools are used to describe the video segment on the right: TextAnnotation datatype, SpatioTemporalMask D, and other visual Ds. A relation between two moving regions is described using the TemporalRelation CS.

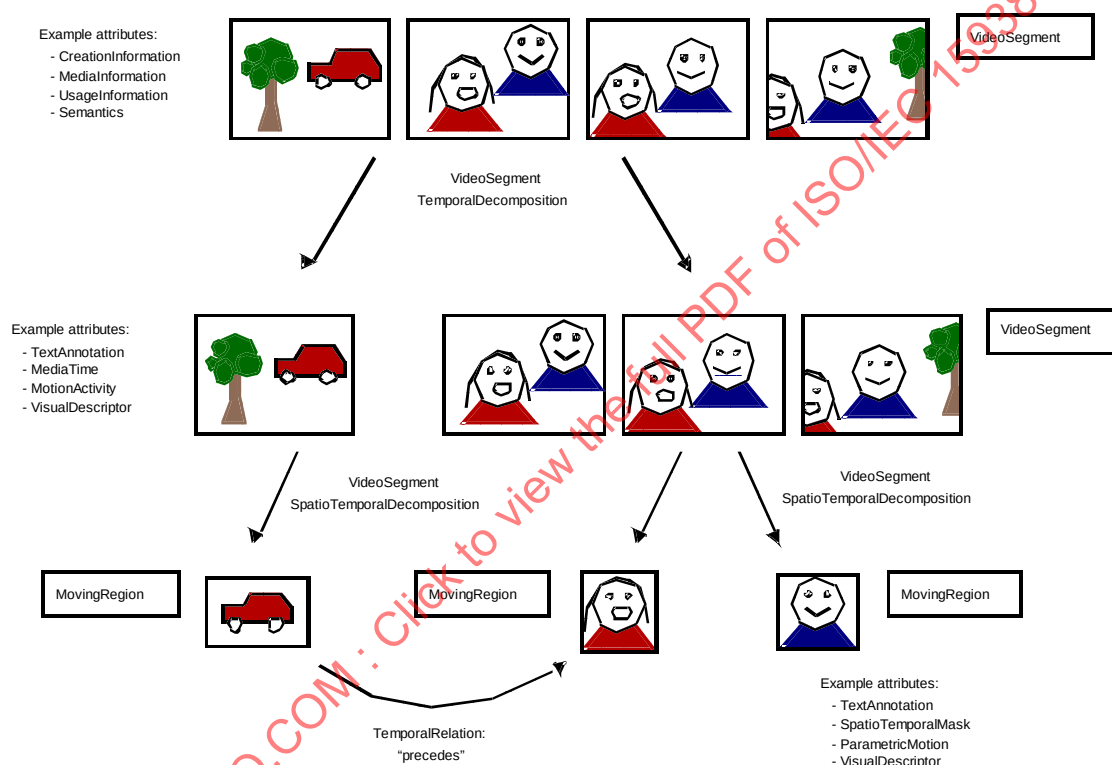


Figure 25 — Illustration of the description of a video segment using tools for describing the structure of multimedia content

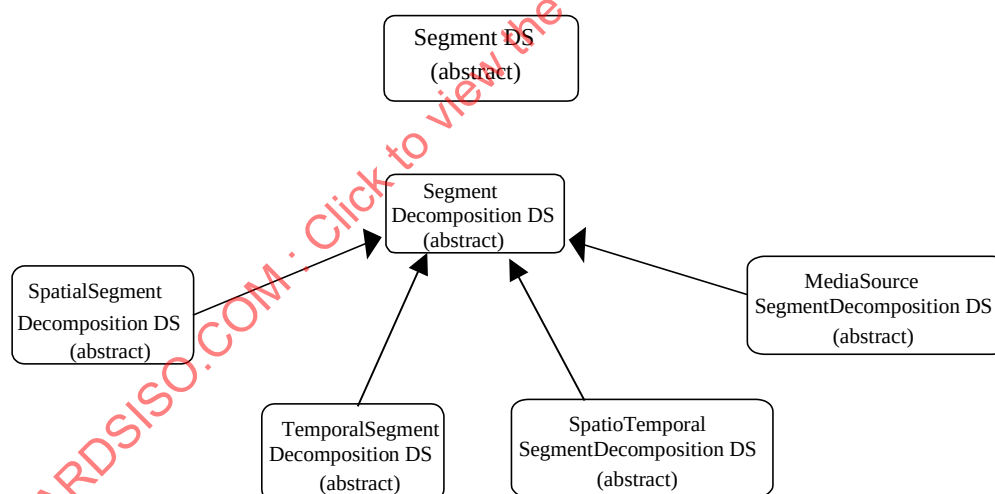
11.2 Base segment description tools

11.2.1 Introduction

This subclause specifies abstract tools that provide the base tools for describing segments of multimedia content and segment decompositions. The following table summarizes the tools specified in this subclause. Figure 26 shows the organization of the base segment description tools. Figure 25 shows an example of the use of the specialized segment description tools for describing a video segment.

Table 16 — Overview of the base segment description tools

<i>Tool</i>	<i>Functionality</i>
Segment DS	This tool is an abstract tool that provides the base of the tools that describe segments of multimedia content. The specialized segment description tools extend the Segment DS.
Base segment decomposition tools	Abstract tools that describe decompositions of segments. The base segment decomposition tools are given as follows: • SegmentDecomposition DS: abstract base of tools that describe decompositions of a segment. • SpatialSegmentDecomposition DS: abstract base of tools that describe spatial decompositions of a segment. • TemporalSegmentDecomposition DS: abstract base of tools that describe temporal decompositions of a segment. • SpatioTemporalSegmentDecomposition DS: abstract base of tools that describe spatio-temporal decompositions of segment. • MediaSourceSegmentDecomposition DS: abstract base of tools that describe media source decompositions of a segment.

**Figure 26 — Overview of the base tools for describing segments of multimedia content and segment decompositions**

11.2.2 Segment DS

11.2.2.1 Introduction

The Segment DS is an abstract tool from which the specialized segment description tools are derived. The Segment DS describes the properties of segments, such as creation information, media information, usage information, semantic information, text annotation, matching hints, point of view descriptions, and so forth.

11.2.2.2 Segment DS syntax

```

<!-- ##### -->
<!-- Definition of Segment DS (11.2.2) -->
<!-- ##### -->

<!-- Definition of Segment DS -->
<complexType name="SegmentType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <choice minOccurs="0">
          <element name="MediaInformation" type="mpeg7:MediaInformationType"/>
          <element name="MediaInformationRef" type="mpeg7:ReferenceType"/>
          <element name="MediaLocator" type="mpeg7:MediaLocatorType"/>
        </choice>
        <element name="StructuralUnit" type="mpeg7:ControlledTermUseType"
          minOccurs="0"/>
        <choice minOccurs="0">
          <element name="CreationInformation"
            type="mpeg7:CreationInformationType"/>
          <element name="CreationInformationRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0">
          <element name="UsageInformation" type="mpeg7:UsageInformationType"/>
          <element name="UsageInformationRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="TextAnnotation" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:TextAnnotationType">
                <attribute name="type" use="optional">
                  <simpleType>
                    <union memberTypes=
                      "mpeg7:termReferenceType string"/>
                  </simpleType>
                </attribute>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Semantic" type="mpeg7:SemanticType"/>
          <element name="SemanticRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="MatchingHint" type="mpeg7:MatchingHintType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="PointOfView" type="mpeg7:PointOfViewType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="Relation" type="mpeg7:RelationType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

11.2.2.3 Segment DS semantics

Semantics of the SegmentType:

<i>Name</i>	<i>Definition</i>
SegmentType	Describes segments of multimedia content (abstract). The specialized segment description tools extend the Segment DS. SegmentType extends DSType.
MediaInformation	Describes the media information related to the segment (optional). MediaInformationType is defined in clause 8.
MediaInformationRef	Describes a reference to the description of media information related to the segment (optional).
MediaLocator	Describes a locator of the segment (optional). This MediaLocator is an alternative to the MediaLocator in MediaInformation, and provides a more direct way of describing media locator when a full description of media information is not needed.
StructuralUnit	Describes the role of the segment in the context of the multimedia content (optional). Examples of values of StructuralUnit include "story", "scene", and "shot". The values may depend on the particular form of the multimedia content, e.g., movie or news report (see Form in Classification DS).
CreationInformation	Describes the creation information related to the segment (optional). CreationInformationType is defined in clause 9.
CreationInformationRef	Describes a reference to the description of creation information related to the segment (optional).
UsageInformation	Describes the usage information related to the segment (optional). UsageInformationType is defined in clause 10.
UsageInformationRef	Describes a reference to the description of usage information related to the segment (optional).
TextAnnotation	Describes text that annotates or describes the content of the segment (optional).
type	Indicates the type or purpose of the textual annotation (optional). Examples of values of type include "scene description", "transcription", and "color description".
Semantic	Describes the semantics of the scene depicted in the segment including the semantics of the real-life or narrative entities depicted in, referenced by, symbolized in, or related to the segment (optional). SemanticType is defined in clause 12.
SemanticRef	Describes a reference to the description of semantic information related to the segment (optional).
MatchingHint	Describes the relative importance of instances of description tools in the Segment DS (or parts of instances of description tools) for matching segments (optional). A segment description can include multiple MatchingHint Ds to describe multiple matching criteria or different combinations of descriptors. MatchingHintType is defined in 11.3.3.
PointOfView	Describes the relative importance of the segment given a specific point of view (optional). PointOfViewType is defined in 11.3.4.
Relation	Describes a relation that the segment participates in (optional). The relations include structural relations defined in 11.10 and possibly other relations.

11.2.2.4 Segment connectivity

Since segments of multimedia content are not always connected or connected in space or time, the composition of the individual connected components that form a segment are described using masks in space and/or time. The TemporalMask D, SpatialMask D, and SpatioTemporalMask D describe masks in time, space, and space and time, respectively, which allow the boundaries and composition of the segments to be described. The different types of connectivity of segments are defined as follows:

- A **temporally connected segment** is a contiguous segment over a temporal interval. Examples include a continuous group of frames from a video, or a temporal sequence of samples of audio.
- A **spatially connected segment** is a contiguous spatial region. An example is a contiguous set of pixels of an image.
- A **spatio-temporally connected segment** is a spatio-temporal segment that (1) appears in a temporally connected segment, and (2) is formed from spatially connected segments at each time instant.

Figure 27 and Figure 28 illustrate examples of segments and their connectivity. Figure 27(a) illustrates a temporally connected segment. Figure 27(b) illustrates a spatially connected segment. Figure 27(c) illustrates a temporally non-connected segment composed of three temporally connected components. Figure 27(d) illustrates a spatially non-connected segment composed of three spatially connected segments.

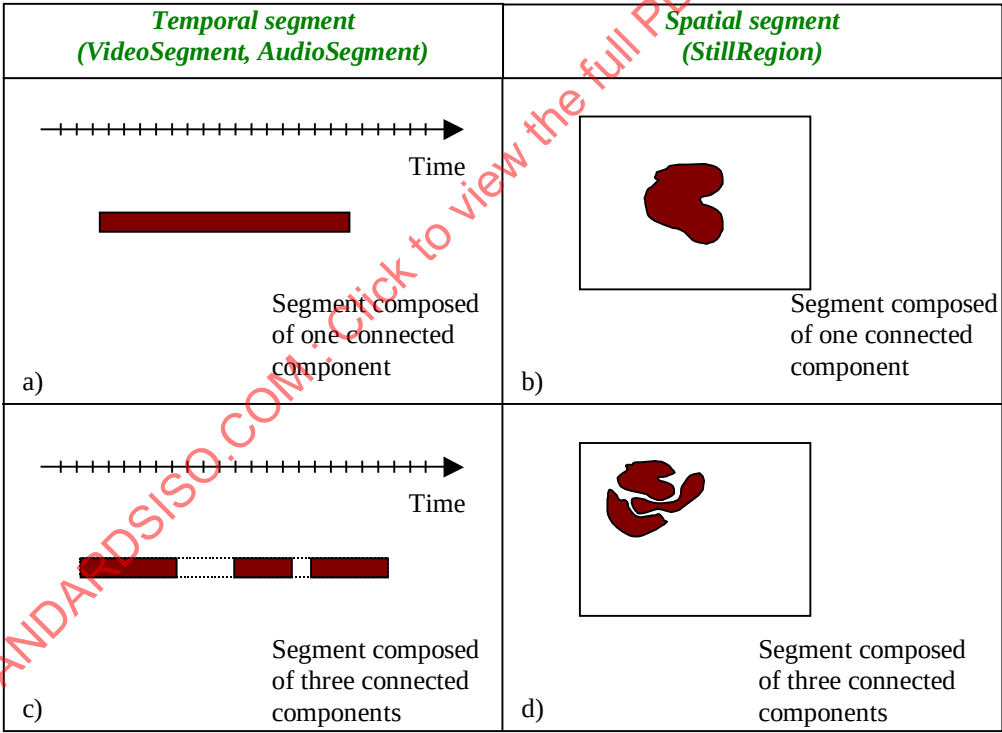


Figure 27 — Examples of the spatial and temporal connectivity of segments: (a) and (b) segments composed of one single connected component; (c) and (d) segments composed of three connected components

Figure 28(a) illustrates a spatio-temporally connected segment. Figure 28(b) illustrates a spatio-temporally non-connected segment.

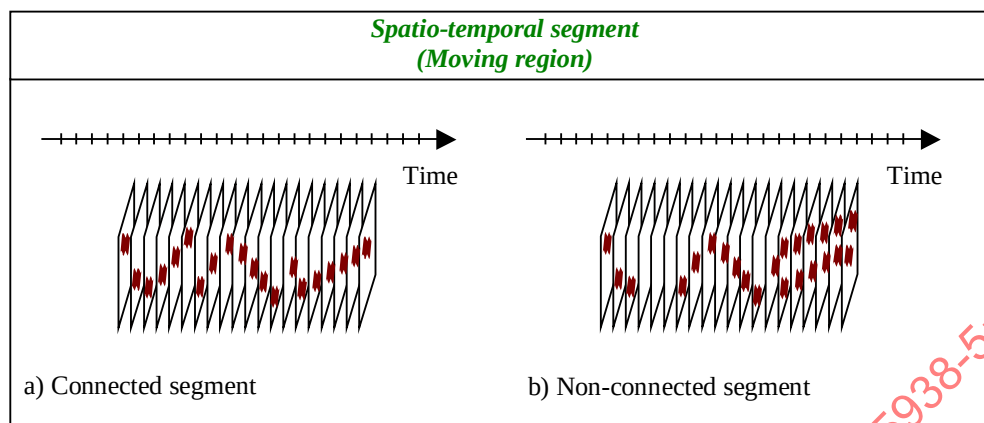


Figure 28 — Examples of the spatio-temporal connectivity of segments: (a) connected segment; (b) non-connected segment

The masks are used to describe the formation and boundaries of the segments. However, the masks do not allow the segment attributes to be applied to the individual components of each mask. In order to describe the attributes of the components of a segment, the segments should be decomposed using the segment decomposition tools in order to allow the Segment DS to be used for describing each sub-segment resulting from the decomposition of the segment.

11.2.3 Base segment decomposition tools

11.2.3.1 Introduction

The SegmentDecomposition DS is an abstract tool forms the base tool for the specific segment decomposition tools that describe decompositions of segments. The SpatialSegmentDecomposition DS, TemporalSegmentDecomposition DS, SpatioTemporalSegmentDecomposition DS, and MediaSourceSegmentDecomposition DS are abstract tools that extend the SegmentDecomposition DS for describing the spatial, temporal, spatio-temporal, and media source decompositions of segments, respectively.

11.2.3.2 Base segment decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of SegmentDecomposition DS (11.2.3) -->
<!-- ##### -->

<!-- Definition of SegmentDecomposition DS -->
<complexType name="SegmentDecompositionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <attribute name="criteria" type="string" use="optional"/>
      <attribute name="overlap" type="boolean" use="optional" default="false"/>
      <attribute name="gap" type="boolean" use="optional" default="false"/>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of SpatialSegmentDecomposition DS -->
<complexType name="SpatialSegmentDecompositionType" abstract="true">
  <complexContent>
```

```

        <extension base="mpeg7:SegmentDecompositionType"/>
    </complexContent>
</complexType>

<!-- Definition of TemporalSegmentDecomposition DS -->
<complexType name="TemporalSegmentDecompositionType" abstract="true">
    <complexContent>
        <extension base="mpeg7:SegmentDecompositionType"/>
    </complexContent>
</complexType>

<!-- Definition of SpatioTemporalSegmentDecomposition DS -->
<complexType name="SpatioTemporalSegmentDecompositionType" abstract="true">
    <complexContent>
        <extension base="mpeg7:SegmentDecompositionType"/>
    </complexContent>
</complexType>

<!-- Definition of MediaSourceSegmentDecomposition DS -->
<complexType name="MediaSourceSegmentDecompositionType" abstract="true">
    <complexContent>
        <extension base="mpeg7:SegmentDecompositionType"/>
    </complexContent>
</complexType>

```

11.2.3.3 Base segment decomposition tools semantics

Semantics of the SegmentDecompositionType:

Name	Definition
SegmentDecompositionType	Describes decompositions of segments (abstract). The specialized segment decomposition tools extend the SegmentDecomposition DS. SegmentDecompositionType extends DSType. A segment decomposition requires that the union of the extents defined by the sub-segments does not extend beyond the extents defined by the parent segment.
criteria	Indicates the criteria used in the segment decomposition (optional). Examples of criteria include "color homogeneity" and "camera motion type."
overlap	Indicates whether or not the segments resulting from the segment decomposition overlap in space and/or time. This attribute value is "false" by default.
gap	Indicates whether or not the segments resulting from the segment decomposition leave gaps with respect to the parent segment. A segment decomposition has gaps if the union of the child sub-segments does not correspond exactly to the parent segment. This attribute value is "false" by default.

Semantics of the SpatialSegmentDecompositionType:

Name	Definition
SpatialSegmentDecompositionType	Abstract type from which the specialized spatial segment decomposition DSs are derived. The SpatialSegmentDecomposition DS describes a spatial decomposition of a segment. SpatialSegmentDecompositionType extends SegmentDecompositionType.

Semantics of the TemporalSegmentDecompositionType:

Name	Definition
TemporalSegmentDecompositionType	Abstract type from which the specialized temporal segment decomposition DSs are derived. The TemporalSegmentDecomposition DS describes a temporal decomposition of a segment. TemporalSegmentDecompositionType extends SegmentDecompositionType.

Semantics of the SpatioTemporalSegmentDecompositionType:

Name	Definition
SpatioTemporalSegmentDecompositionType	Abstract type from which the specialized spatio-temporal segment decomposition DSs are derived. The SpatioTemporalSegmentDecomposition DS describes a spatio-temporal decomposition of a segment. SpatioTemporalSegmentDecompositionType extends SegmentDecompositionType.

Semantics of the MediaSourceSegmentDecompositionType:

Name	Definition
MediaSourceSegmentDecompositionType	Abstract type from which the specialized media source segment decomposition DSs are derived. The MediaSourceSegmentDecomposition DS describes a media source decomposition of a segment. MediaSourceSegmentDecompositionType extends SegmentDecompositionType.

11.2.3.4 Segment decomposition type

11.2.3.4.1 Introduction

The segment decomposition tools describe decompositions of segments. The segment decompositions can be recursive and form a tree hierarchy of segments. For example, a segment tree description of a video can be

produced to describe the hierarchical organization of the video into scenes, shots, and sub-shots. Similar decompositions can be formed temporally for audio, spatially for images, and spatio-temporally for video. Specialized segment decomposition tools describe the decomposition of a segment into constituent media tracks or media sources. For example, the media decompositions can apply to AV content containing multiple audio tracks for stereo surround sound or multiple video tracks for different camera viewpoints.

The segment decomposition tools can also describe directed acyclic graph types of segment decompositions in which the same child segment has multiple parent segments. The child segment description may be referenced from the multiple descriptions of segment decomposition tools.

11.2.3.4.2 Overlap and gap in segment decomposition

The SegmentDecomposition DS requires that the extent of the union of the children segments does not extend beyond the extent of the parent segment. For example, in the temporal decomposition of the parent segment in Figure 29(a) into three child segments, the children segments do not extend beyond the temporal extent of the parent segment. However, the decomposition of a segment may have gaps or overlaps between the segments resulting from the decomposition as follows:

- No gaps, no overlaps (partition in the mathematical sense)
- No gaps, overlaps
- Gaps, no overlaps
- Gaps, overlaps

Figure 29 shows examples of decompositions of temporal segments that illustrate the concepts of gaps and overlaps. Figure 29(a) and Figure 29(b) depict decompositions that do not have gaps or overlaps. In both cases the union of the extents of the child sub-segments corresponds exactly to the extent of the parent segment and the intersection among any pair of child sub-segments is empty. This applies also in this case shown in Figure 29(b) in which the parent is itself not connected. Figure 29(c) shows a decomposition with gaps but no overlaps. Finally, Figure 29(d) illustrates a more complex case where the parent is composed of two connected components and is decomposed with gaps and overlaps in three children: the first one is itself composed of two connected components, while the two remaining children are composed of a single connected component. These examples also demonstrate the fact that segment connectivity is independent of the segment decomposition: a non-connected segment can be decomposed with no gaps and a connected segment can be decomposed with gaps.

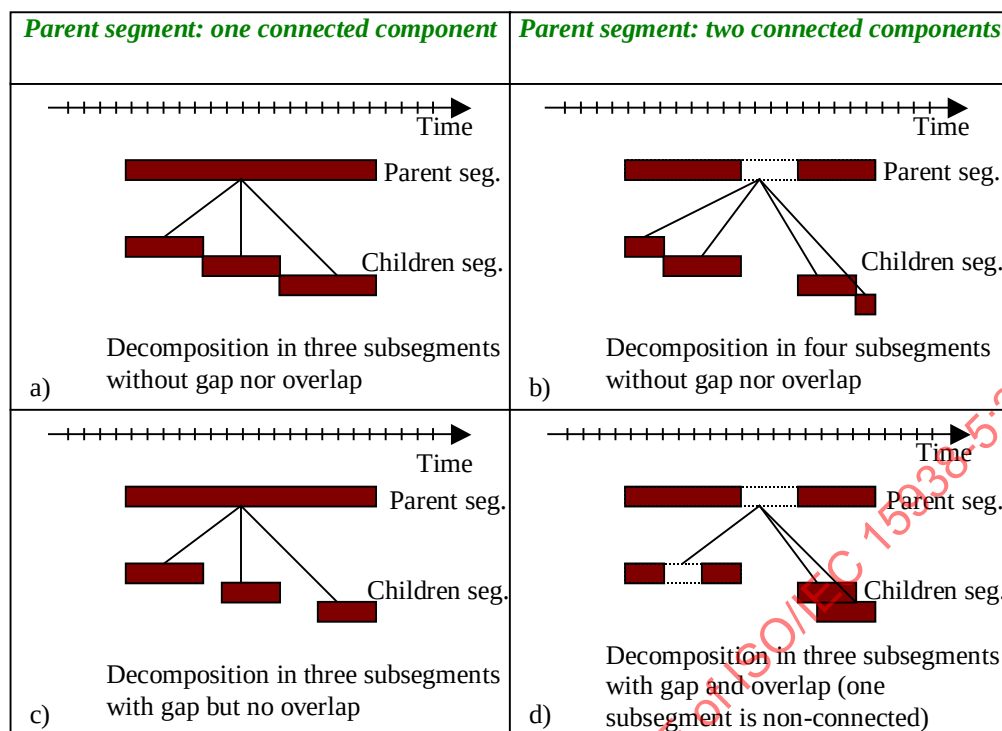


Figure 29 — Examples of temporal segment decompositions: a) and b) segment decompositions with neither gaps nor overlaps; c) and d) segment decompositions with gaps or overlaps

11.2.3.4.3 Segment decomposition constraints

The applicability of the specialized segment decompositions to the different types of segments is limited as shown in Figure 30. Each Segment DS in the figure represents itself and the types extended from it, except where defined differently. For example, as shown in Figure 30, a still region can only be decomposed in space into other still regions including image texts. However, the Mosaic DS is a special case of StillRegion DS that cannot result from any segment decomposition. In Figure 30, the following notation is used: "time" = temporal decomposition; "space" = spatial decomposition; "spaceTime" = spatio-temporal decomposition; and "media" = media source decomposition.

Note: a segment may be decomposed into one or more sub-segments of different types, e.g. a video segment may be decomposed into still regions and moving regions, which may be decomposed into still regions at the same time. However, all the combinations of parent segment type, segment decomposition type, and child segment type are not valid. For example, an audio segment can only be decomposed in time or media into other audio segments; the spatial decomposition of an audio segment into still regions is not valid.

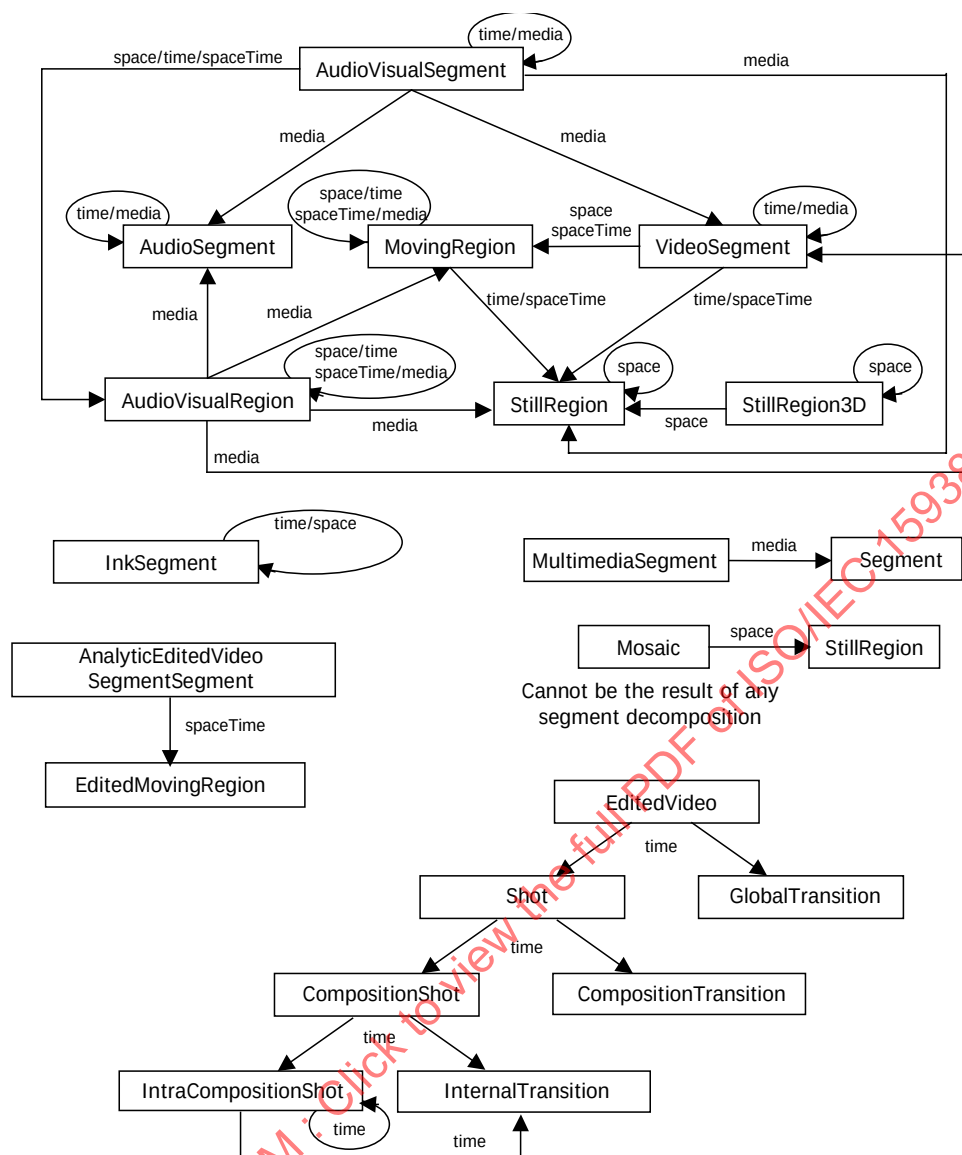


Figure 30 — Overview of the valid types of segment decompositions among the specialized Segment DSs

11.3 Segment attribute description tools

11.3.1 Introduction

This subclause specifies tools for describing attributes of multimedia segments including masks, matching hints, and point of view descriptions. The following table summarizes the segment attribute description tools:

Table 17 — Overview of the segment attribute description tools

Tool	Functionality
Segment mask description tools	Describe masks of multimedia data (images, video, audio, scenes, byte streams). These tools apply to the specialized segment description tools.
MatchingHint D	Describes the relative importance of instances of audio and visual description tools (or parts of instances of audio and visual description tools) in a Segment DS

<i>Tool</i>	<i>Functionality</i>
	description for matching. This tool applies to the Segment DS.
PointOfView D	This tool describes the relative importance of segments given a specific viewpoint. This tool applies to the Segment DS.

11.3.2 Segment mask description tools

11.3.2.1 Introduction

This subclause specifies tools for describing masks of multimedia data (images, video, audio, scenes, byte streams). Figure 35 shows the organization of the mask description tools, which are summarized Table 18.

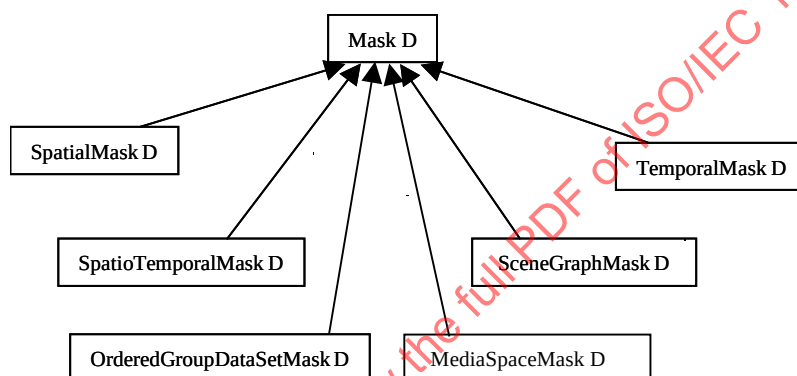


Figure 31 — Organization of segment mask description tools

Table 18 — Overview of the mask description tools

<i>Tool</i>	<i>Functionality</i>
Mask D	This tool describes masks of multimedia data. MaskType is the abstract base type for the specialized mask types.
SpatialMask D	This tool describes a mask in 2-D space. The SpatialMask D can be used to describe the spatial boundaries of regions in images or video frames.
TemporalMask D	This tool describes a mask in time. The TemporalMask D can be used to describe temporal interval information of segments of audio or video.
SpatioTemporalMask D	This tool describes a mask in space and time. The SpatioTemporalMask D can be used to describe together the spatio-temporal boundary information of regions in a video sequence.
MediaSpaceMask D	This tool describes a mask as a sequence of bytes. The MediaSpaceMask D can be used to describe a set of byte-ranges from a byte-stream representation of multimedia data.
SceneGraphMask D	This tool describes a mask of sub-graphs in a scene graph. The SceneGraphMask D can be used to describe a set of sub-graphs from a scene graph representation of multimedia data.

OrderedGroupDataSetMask D	This tool describes a mask from an ordered group dataset. The OrderedGroupDataSetMask D can be used to describe a set of nodes or other units from an ordered group dataset of multimedia data.

11.3.2.2 Mask D

11.3.2.2.1 Introduction

The Mask D describes a mask of multimedia data. MaskType is the abstract base type for the specialized mask types.

11.3.2.2.2 Mask D syntax

```
<!-- ##### -->
<!-- Definition of Mask D (11.3.2.2) -->
<!-- ##### -->

<!-- Definition of Mask D -->
<complexType name="MaskType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DType"/>
  </complexContent>
</complexType>
```

11.3.2.2.3 Mask D semantics

Semantics of the MaskType:

Name	Definition
MaskType	Identifies the boundaries of sub-regions, sub-intervals, and other kinds of units of multimedia content that can be identified by applying a mask to the multimedia data (abstract). MaskType extends DType.

11.3.2.3 SpatialMask D

11.3.2.3.1 Introduction

The SpatialMask D describes a mask in 2-D space. The spatial mask is comprised of an unbounded set of sub-regions, where each sub-region is described using the RegionLocator D (defined in ISO/IEC 15398-3). The SpatialMask D is used by the StillRegion DS to describe the boundary of a region within an image or video frame using a bounding box or polygon (see 11.4.2).

11.3.2.3.2 SpatialMask D syntax

```
<!-- ##### -->
<!-- Definition of SpatialMask D (11.3.2.3) -->
<!-- ##### -->

<!-- Definition of SpatialMask D -->
<complexType name="SpatialMaskType">
  <complexContent>
```

```

    <extension base="mpeg7:MaskType">
      <sequence>
        <element name="SubRegion" type="mpeg7:RegionLocatorType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

11.3.2.3.3 SpatialMask D semantics

Semantics of the SpatialMaskType:

Name	Definition
SpatialMaskType	Describes a mask in 2-D space, which can be used to describe the spatial boundaries of regions in images or video frames. SpatialMaskType extends MaskType.
SubRegion	Describes the unbounded set of sub-regions that comprise the mask. Each sub-region is described using RegionLocatorType (defined in ISO/IEC 15398-3).

11.3.2.3.4 SpatialMask D examples (informative)

The following example illustrates the use of the SpatialMask D for describing the spatial mask shown in Figure 27(d), which consists of three spatially connected sub-regions. Each sub-region is described using a polygon that denotes its boundary.

```

<SpatialMask>
  <SubRegion>
    <Polygon>
      <Coords mpeg7:dim="2 5"> 5 25 10 20 15 15 10 10 5 15 </Coords>
    </Polygon>
  </SubRegion>
  <SubRegion>
    <Polygon>
      <Coords mpeg7:dim="2 6"> 7 24 15 24 20 27 18 25 15 22 7 22 </Coords>
    </Polygon>
  </SubRegion>
  <SubRegion>
    <Polygon>
      <Coords mpeg7:dim="2 4"> 7 30 15 30 15 25 7 25</Coords>
    </Polygon>
  </SubRegion>
</SpatialMask>

```

11.3.2.4 TemporalMask D

11.3.2.4.1 Introduction

The TemporalMask D describes a mask in time. The temporal mask is comprised of an unbounded set of sub-intervals, where each sub-interval is described using the MediaTime datatype, which describes a media time point or interval.

11.3.2.4.2 TemporalMask D syntax

```
<!-- ##### -->
<!-- Definition of TemporalMask D (11.3.2.4) -->
<!-- ##### -->

<!-- Definition of TemporalMask D -->
<complexType name="TemporalMaskType">
  <complexContent>
    <extension base="mpeg7:MaskType">
      <sequence>
        <element name="SubInterval" type="mpeg7:MediaTimeType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.3.2.4.3 TemporalMask D semantics

Semantics of the TemporalMaskType:

Name	Definition
TemporalMaskType	Describes a mask in time, which can be used to describe the temporal boundaries of segments in audio and video data. TemporalMaskType extends MaskType.
SubInterval	Describes the unbounded set of sub-intervals, where each sub-interval is described using MediaTimeType, which describes a media time point or interval.

11.3.2.4.4 TemporalMask D examples (informative)

The following example illustrates the use of the TemporalMask D for describing the temporal mask shown in Figure 29(b), which consists of two temporally connected sub-intervals. Each sub-interval is described using a media time description (time point and duration) that denotes its boundaries in time.

```
<TemporalMask>
  <SubInterval>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT6M</MediaDuration>
  </SubInterval>
  <SubInterval>
    <MediaTimePoint>T00:07:00</MediaTimePoint>
    <MediaDuration>PT3M</MediaDuration>
  </SubInterval>
</TemporalMask>
```

11.3.2.5 SpatioTemporalMask D

11.3.2.5.1 Introduction

The SpatioTemporalMask D describes a mask in space and time. The spatio-temporal mask is comprised of an unbounded set of sub-regions, where each sub-region is described using the SpatioTemporalLocator D

(defined in ISO/IEC 15938-3). The SpatioTemporalMask D is used by the MovingRegion DS to describe the spatio-temporal boundary of a moving region within a video sequence (see 11.4.10).

11.3.2.5.2 SpatioTemporalMask D syntax

```
<!-- ##### -->
<!-- Definition of SpatioTemporalMask D (11.3.2.5) -->
<!-- ##### -->

<!-- Definition of SpatioTemporalMask D -->
<complexType name="SpatioTemporalMaskType">
  <complexContent>
    <extension base="mpeg7:MaskType">
      <sequence>
        <element name="SubRegion" type="mpeg7:SpatioTemporalLocatorType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.3.2.5.3 SpatioTemporalMask D semantics

Semantics of the SpatioTemporalMaskType:

Name	Definition
SpatioTemporalMaskType	Describes a mask in space and time, which can be used to describe together the spatio-temporal boundary information of regions in a video sequence. SpatioTemporalMaskType extends MaskType.
SubRegion	Describes the unbounded set of sub-regions that comprise the spatio-temporal mask. Each sub-region is described using SpatioTemporalLocatorType (defined in ISO/IEC 15938-3).

11.3.2.5.4 SpatioTemporalMask D examples (informative)

The following example illustrates the use of the SpatioTemporalMask D for describing a spatio-temporal mask comprised of two spatio-temporal sub-regions. Each sub-region is described by trajectories of the representative points of a reference region using a spatio-temporal locator description.

```
<SpatioTemporalMask>
  <SubRegion>
    <FigureTrajectory type="rectangle">
      <MediaTime>
        <MediaTimePoint>T00:00:15</MediaTimePoint>
        <MediaDuration>PT1M15S</MediaDuration>
      </MediaTime>
      <Vertex><!-- more elements here --></Vertex>
      <Vertex><!-- more elements here --></Vertex>
      <Vertex><!-- more elements here --></Vertex>
      <!-- more elements here -->
    </FigureTrajectory>
  </SubRegion>
  <SubRegion>
```

```

    <FigureTrajectory type="ellipse">
      <MediaTime>
        <MediaTimePoint>T00:14:00</MediaTimePoint>
        <MediaDuration>PT1M7S</MediaDuration>
      </MediaTime>
      <Vertex><!-- more elements here --></Vertex>
      <Vertex><!-- more elements here --></Vertex>
      <Vertex><!-- more elements here --></Vertex>
      <!-- more elements here -->
    </FigureTrajectory>
  </SubRegion>
</SpatioTemporalMask>

```

11.3.2.6 MediaSpaceMask D

11.3.2.6.1 Introduction

The MediaSpaceMask D describes a mask as a sequence of bytes. The MediaSpaceMask D can be used to describe a set of byte-ranges from a byte-stream representation of multimedia data. The MediaSpaceMask D allows a mask to be described by pointing and segmenting the media data directly without parsing the content.

11.3.2.6.2 MediaSpaceMask D syntax

```

<!-- ##### -->
<!-- Definition of MediaSpaceMask D (11.3.2.6) -->
<!-- ##### -->

<!-- Definition of MediaSpaceMask D -->
<complexType name="MediaSpaceMaskType">
  <complexContent>
    <extension base="mpeg7:MaskType">
      <sequence>
        <element name="SubInterval" minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <attribute name="offset"
              type="nonNegativeInteger" use="required"/>
            <attribute name="length" type="positiveInteger" use="optional"/>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

11.3.2.6.3 MediaSpaceMask D semantics

Semantics of the MediaSpaceMaskType:

Name	Definition
MediaSpaceMaskType	Describes a mask as a sequence of bytes, which can be used to describe a set of byte-ranges from a byte-stream representation of multimedia data. MediaSpaceMaskType extends MaskType.
SubInterval	Describes a set of sub-intervals of bytes by describing the byte offset and byte length of each sub-interval.

Name	Definition
offset	Indicates the byte offset of the sub-interval. The byte offset specifies the number of bytes to be skipped from the beginning of the byte stream in order to reach the start of the sub-interval.
length	Indicates the length of the sub-interval in number of bytes (optional). If the byte length is not specified, the end of the sub-interval is the end of the byte stream.

11.3.2.6.4 MediaSpaceMask D examples (informative)

The following example illustrates the use of the MediaSpaceMask D for describing a media space mask consisting of two sub-intervals from the media stream shown in Figure 32.

```
<MediaSpaceMask>
  <SubInterval offset="105" length="34"/>
  <SubInterval offset="167" length="50"/>
</MediaSpaceMask>
```

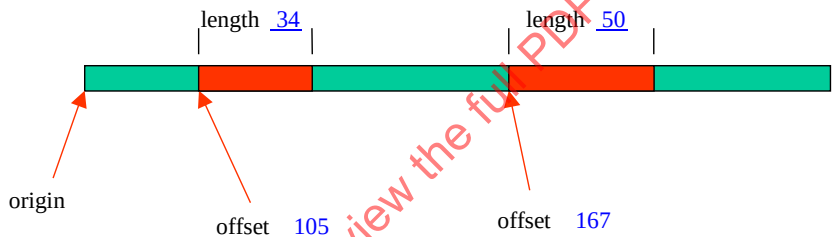


Figure 32 — Example of a media space mask

11.3.2.7 SceneGraphMask D

11.3.2.7.1 Introduction

The SceneGraphMask D describes a mask of sub-graphs in a scene graph. The scene graph mask is comprised of an unbounded set of sub-graphs, which can be identified by reference or sub-graph number. The SceneGraphMask D can be used to describe a set of sub-graphs from a scene graph representation of multimedia data.

11.3.2.7.2 SceneGraphMask D syntax

```
<!-- ##### -->
<!-- Definition of SceneGraphMask D (11.3.2.7) -->
<!-- ##### -->

<!-- Definition of SceneGraphMask D -->
<complexType name="SceneGraphMaskType">
  <complexContent>
    <extension base="mpeg7:MaskType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="SubGraph" type="anyURI"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="SubGraphNum" type="nonNegativeInteger"/>
</choice>
</extension>
</complexContent>
</complexType>
```

11.3.2.7.3 SceneGraphMask D semantics

Semantics of the SceneGraphMaskType:

Name	Definition
SceneGraphMaskType	Describes a mask of sub-graphs in a scene graph, which can be used to describe a set of sub-graphs from a scene graph representation of multimedia data. SceneGraphMaskType extends MaskType.
SubGraph	Describes a set of sub-graphs in the scene graph mask by pointing to the URI address of each sub-graph's root node.
SubGraphNum	Identifies the scene graph number of a sub-graph's root node. The sub-graphs are numbered in the scene graph starting at 0 for the root node and traversing in bread-first ordering.

11.3.2.7.4 SceneGraphMask D examples (informative)

The following example illustrates the use of the SceneGraphMask D for describing a scene graph mask comprised of two sub-graphs (#1 and #21) from the scene graph shown in Figure 33.

```
<SceneGraphMask>
  <SubGraphNum>1</SubGraphNum>
  <SubGraphNum>21</SubGraphNum>
</SceneGraphMask>
```

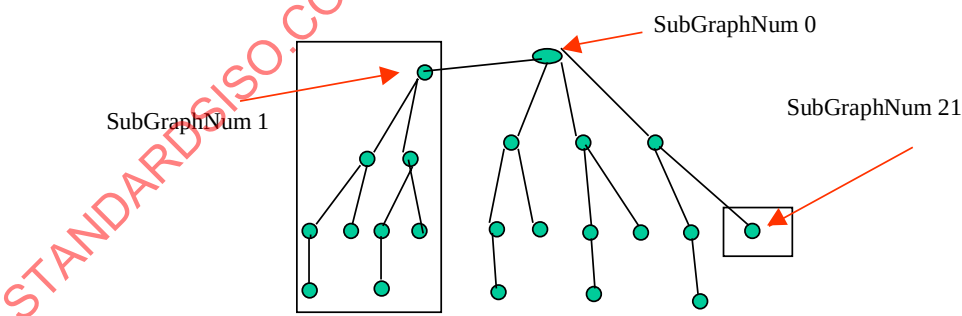


Figure 33 — Example of a scene graph mask

11.3.2.8 OrderedGroupDataSetMask D

11.3.2.8.1 Introduction

The OrderedGroupDataSetMask D describes a mask of sub-intervals in an ordered group dataset. The ordered group data set mask is comprised of an unbounded set of sub-intervals, each of which identifies a component layer and unit of content. The OrderedGroupDataSetMask D can be used to describe a mask of content that is organized as an ordered grouping data structure, such as ink content or an MPEG4/VRML scene graph.

For example, ink content can be organized into an ordered grouping data structure in which the component layer gives the strokes, and the unit layer gives the ink data points within each stroke. For MPEG4 or VRML content, each sub-graph can be described as a set, and each leaf node under the sub-graph can be described as a component. As a result, the startComponent and endComponent attributes can be used to locate a subset of leaf nodes under a sub-graph.

11.3.2.8.2 OrderedGroupDataSetMask D syntax

```
<!-- ##### -->
<!-- Definition of OrderedGroupDataSetMask D (11.3.2.8) -->
<!-- ##### -->

<!-- Definition of OrderedGroupDataSetMask D -->
<complexType name="OrderedGroupDataSetMaskType">
  <complexContent>
    <extension base="mpeg7:MaskType">
      <sequence>
        <element name="SubInterval" minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <attribute name="setRef" type="anyURI" use="optional"/>
            <attribute name="startComponent" type="nonNegativeInteger"
              use="optional"/>
            <attribute name="startUnit" type="nonNegativeInteger"
              use="optional"/>
            <attribute name="endComponent" type="nonNegativeInteger"
              use="optional"/>
            <attribute name="endUnit" type="nonNegativeInteger"
              use="optional"/>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.3.2.8.3 OrderedGroupDataSetMask D semantics

Semantics of the OrderedGroupDataSetMaskType:

Name	Definition
OrderedGroupDataSetMaskType	Describes a mask of sub-intervals in an ordered group dataset, which can be used to describe a mask of content that is organized as an ordered grouping data structure, such as ink content or an MPEG4/VRML scene graph. OrderedGroupDataSetMaskType extends MaskType.

SubInterval	Describes a sub-interval in an ordered group data set. A single sub-interval shall not span multiple sets from the ordered group data set.
setRef	Describes a reference to the ordered group data set where the sub-interval is defined and from where the units and components are taken (optional).
startComponent	Indicates the number of the beginning component in the sub-interval (optional). If absent, the component is the first one in the data set.
startUnit	Indicates the number of the starting unit of the beginning component in the sub-interval (optional). If absent, the unit is the first one in the start component.
endComponent	Indicates the number of the end component in the sub-interval (optional). If absent, the component is the last one in the data set.
endUnit	Indicate the number of the end unit of the end component in the sub-interval (optional). If absent, the unit is the last one in the end component.

11.3.2.8.4 OrderedGroupDataSetMask D example (informative)

The following example illustrates the use of the OrderedGroupDataSetMask D for describing an ordered group data set mask consisting of two sub-intervals as shown in Figure 34. This example organizes ink content by grouping ink strokes into ordered group data sets. Each ordered group data set consists of ink strokes to form the component layer, and each stroke consists of ink data points to form the unit layer.

```
<OrderedGroupDataSetMask>
  <SubInterval setRef="http://www.../demo.stk#1" startComponent="0" startUnit="0"
    endComponent="0" endUnit="U"/>
  <SubInterval setRef="http://www.../demo.stk#Set_N" startComponent="1" startUnit="0"
    endComponent="1" endUnit="V"/>
</OrderedGroupDataSetMask>
```

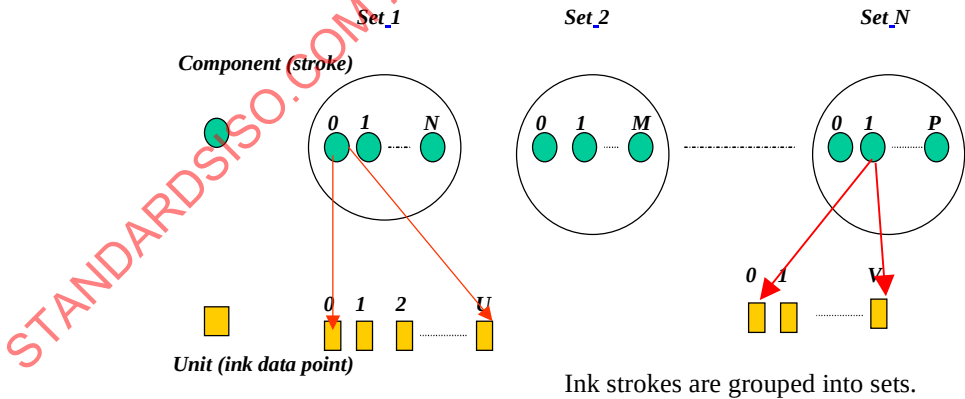


Figure 34 — Example of an ordered group mask

11.3.3 MatchingHint D

11.3.3.1 Introduction

The MatchingHint D describes the relative importance of instances of audio and visual description tools (e.g, the color structure of a still region) or parts of instances of audio and visual description tools (e.g, the fourth bin of the color structure of a still region) in Segment DS descriptions for matching. A part of a description tool is any combination of the attributes and elements of the description tool. The MatchingHint D can improve the performance of a retrieval application based on feature similarity because the most relevant descriptors for matching may vary from segment to segment. The MatchingHint D applies to the Segment DS.

11.3.3.2 MatchingHint D syntax

```
<!-- ##### -->
<!-- Definition of MatchingHint D (11.3.3) -->
<!-- ##### -->

<!-- Definition of MatchingHint D -->
<complexType name="MatchingHintType">
  <complexContent>
    <extension base="mpeg7:DType">
      <sequence>
        <element name="Hint" minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="mpeg7:ReferenceType">
                <attribute name="value" type="mpeg7:zeroToOneType"
                  use="required"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
      <attribute name="reliability" type="mpeg7:zeroToOneType" use="optional"
        default="1.0"/>
    </extension>
  </complexContent>
</complexType>
```

11.3.3.3 MatchingHint D semantics

Semantics of the MatchingHintType:

Name	Definition
MatchingHintType	Describes the relative importance of instances of audio and visual description tools (or parts of instances of audio and visual description tools) in Segment DS descriptions for matching. A part of a description tool is an attribute or an element of the description tool. MatchingHintType extends DType.
Hint	References an instance of an audio or visual descriptor or description scheme (or a part of an audio or visual description tool) of a Segment DS description and assigns it a relative importance value.
value	Indicates the relative importance value. A value of 1.0 signifies the highest importance and a value of 0.0 signifies the lowest importance.

Name	Definition
reliability	Indicates the reliability of the importance values in allowing effective matching. For example, if the reliability is high (close to 1.0), the matching hint is effective in improving the matching results. On the other hand, if the reliability is low (close to 0.0), the matching hint does not reliably improve the matching results. This attribute value is 1.0 by default.

11.3.3.4 MatchingHint D examples (informative)

The following example illustrates the use of the MatchingHint D for describing relative importance values to the several instances of visual description tools in a StillRegion DS description. The MatchingHint D descriptions can be generated and updated continuously by user's feedback. The reliability of the importance values may vary depending on how the values have been updated. For example, matching hint values learned from a large number of feedback sessions from reliable experts may have higher reliability.

```
<StillRegion id="IMG0001">
  <MatchingHint reliability="0.900000">
    <Hint value="0.455" xpath="../../../ColorStructure"/>
    <Hint value="0.265" xpath="../../../DominantColor"/>
    <Hint value="0.280" xpath="../../../HomogeneousTexture"/>
  </MatchingHint>

  <VisualDescriptor xsi:type="DominantColorType">
    <!-- more elements here -->
  </VisualDescriptor>
  <VisualDescriptor xsi:type="ColorStructureType">
    <!-- more elements here -->
  </VisualDescriptor>
  <VisualDescriptor xsi:type="HomogeneousTextureType">
    <!-- more elements here -->
  </VisualDescriptor>
  <!-- more elements here -->
</StillRegion>
```

11.3.4 PointOfView D

11.3.4.1 Introduction

The PointOfView D describes the relative importance of segments given a specific viewpoint. The PointOfView D applies to the Segment DS.

11.3.4.2 PointOfView D syntax

```
<!-- ##### -->
<!-- Definition of PointOfView D (11.3.4) -->
<!-- ##### -->

<!-- Definition of PointOfView D -->
<complexType name="PointOfViewType">
  <complexContent>
    <extension base="mpeg7:DType">
      <sequence>
        <element name="SupplementalInformation" type="mpeg7:TextAnnotationType"
          minOccurs="0"/>
        <element name="Importance" minOccurs="1" maxOccurs="unbounded">
```



```

        <complexType>
          <sequence>
            <element name="Value" type="mpeg7:zeroToOneType"/>
            <element name="SegmentRef" type="mpeg7:ReferenceType"
              minOccurs="0"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
    <attribute name="viewpoint" type="string" use="required"/>
  </extension>
</complexContent>
</complexType>

```

11.3.4.3 PointOfView D semantics

Semantics of the PointOfViewType:

Name	Definition
PointOfViewType	Describes the relative importance of one or more segments given a specific viewpoint. PointOfViewType extends DType.
SupplementalInformation	Describes the supplementary information for the management of PointOfView D description such as creator, creation date, provider, and so on (optional).
Importance	Describes the relative importance value and the references to a Segment DS description to be assigned a relative importance value.
Value	Indicates the relative importance value. A value of 1.0 signifies the highest importance. The relative importance values within the same scope of a PointOfView D description need not be normalized to add to one.
SegmentRef	References a Segment DS description to be assigned a relative importance value (optional). If the reference to a Segment DS description is absent, the relative importance value is assigned to the Segment DS description that contains the PointOfView D description.
viewpoint	Indicates the viewpoint for weighting the segments. The value of the viewpoint attribute describes the content of the segment. Examples of viewpoints for the video segments of a football game are "Team A" and "Team B".

The PointOfView D can be used in two different forms: the nested form and the separated form. In the nested form, the Segment DS description includes the PointOfView D description with the importance value for a given viewpoint. In the separated form, the PointOfView D description stands apart from the Segment DS description; the PointOfView D description points to the Segment DS description using the SegmentRef element. Informative examples of both types of descriptions of the PointOfView D are provided in the following subclause.

The scope of a PointOfView D description determines the set of importance values that can be compared to each other.

Note: the PointOfView D attaches relative importance values to segments for a given viewpoint. The scope of a separated-form description of the PointOfView D is one PointOfView D description. In the nested form, the scope of a PointOfView D description is limited to the PointOfView D descriptions with the same viewpoint included within the siblings of the parent Segment DS description.

11.3.4.4 PointOfView D examples (informative)

The following examples illustrate the use of the PointOfView D for describing the relative importance of AV segments given a specific viewpoint. In these examples, the AV segments belong to the video of a football game between Team A and Team B. The PointOfView D describes the viewpoints of "Team A" and "Team B". In this example, it is assumed that the content creator assigned the relative importance values to the AV segments manually using an authoring tool. The examples show the description of the PointOfView D in the nested form and in the separated form, respectively.

In the first example, the description contains several PointOfView D descriptions with the same viewpoint "Team A". The importance values of the PointOfView D descriptions with viewpoint "Team A" included within the AudioVisualSegment DS descriptions with ids "seg1", "seg2", ..., "seg20", ... can be compared; the same is true within the AudioVisualSegment DS descriptions with ids of "seg1_2", "seg2_2", ..., "seg20_2", ... However, the importance values of PointOfView D descriptions across the two sets of AudioVisualSegment DS descriptions cannot be compared because they are of different scope.

```
<!-- Description of PointOfView D: nested form -->
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioVisualType">
      <AudioVisual>
        <TemporalDecomposition>
          <AudioVisualSegment id="FootBallGame">
            <TemporalDecomposition>
              <AudioVisualSegment id="seg1">
                <PointOfView viewpoint="Team A">
                  <Importance>
                    <Value>0.3</Value>
                  </Importance>
                </PointOfView>
                <PointOfView viewpoint="Team B">
                  <Importance>
                    <Value>0.7</Value>
                  </Importance>
                </PointOfView>
              </AudioVisualSegment>
              <AudioVisualSegment id="seg2">
                <PointOfView viewpoint="Team A">
                  <Importance>
                    <Value>0.5</Value>
                  </Importance>
                </PointOfView>
              </AudioVisualSegment>
              <!-- more elements here -->
              <AudioVisualSegment id="seg20">
                <PointOfView viewpoint="Team A">
                  <Importance>
                    <Value>0.8</Value>
                  </Importance>
                </PointOfView>
                <PointOfView viewpoint="Team B">
                  <Importance>
                    <Value>0.2</Value>
                  </Importance>
                </PointOfView>
              </AudioVisualSegment>
              <!-- more elements here -->
            </TemporalDecomposition>
          </TemporalDecomposition>
          <AudioVisualSegment id="seg1_2">
```

```

        <PointOfView viewpoint="Team A">
            <Importance>
                <Value>0.3</Value>
            </Importance>
        </PointOfView>
    </AudioVisualSegment>
    <AudioVisualSegment id="seg2_2">
        <PointOfView viewpoint="Team A">
            <Importance>
                <Value>0.5</Value>
            </Importance>
        </PointOfView>
    </AudioVisualSegment>
    <!-- more elements here -->
    <AudioVisualSegment id="seg20_2">
        <PointOfView viewpoint="Team A">
            <Importance>
                <Value>0.8</Value>
            </Importance>
        </PointOfView>
    </AudioVisualSegment>
    <!-- more elements here -->
</TemporalDecomposition>
</AudioVisualSegment>
</TemporalDecomposition>
</AudioVisual>
</MultimediaContent>
</Description>
</Mpeg7>

```

In the second example, the scope of each PointOfView D description is each description itself.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioVisualType">
      <AudioVisual>
        <TemporalDecomposition>
          <AudioVisualSegment id="FootballGame">
            <!-- Description of PointOfView D: separated form -->
            <PointOfView viewpoint="Team A">
              <Importance>
                <Value>0.3</Value>
                <SegmentRef href="#seg1"/>
              </Importance>
              <Importance>
                <Value>0.5</Value>
                <SegmentRef href="#seg2"/>
              </Importance>
              <!-- more elements here -->
              <Importance>
                <Value>0.8</Value>
                <SegmentRef href="#seg20"/>
              </Importance>
              <!-- more elements here -->
            </PointOfView>
            <PointOfView viewpoint="Team B">
              <Importance>
                <Value>0.7</Value>
                <SegmentRef href="#seg1"/>
              </Importance>
            </PointOfView>
          </AudioVisualSegment>
        </TemporalDecomposition>
      </AudioVisual>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

```

        </Importance>
        <!-- more elements here -->
        <Importance>
            <Value>0.2</Value>
            <SegmentRef href="#seg20"/>
        </Importance>
        <!-- more elements here -->
    </PointOfView>
    <!-- Description of AV segment tree -->
    <TemporalDecomposition>
        <AudioVisualSegment id="seg1">
            <!-- more elements here -->
        </AudioVisualSegment>
        <AudioVisualSegment id="seg2">
            <!-- more elements here -->
        </AudioVisualSegment>
        <!-- more elements here -->
        <AudioVisualSegment id="seg20">
            <!-- more elements here -->
        </AudioVisualSegment>
        <!-- more elements here -->
    </TemporalDecomposition>
    </AudioVisualSegment>
    </TemporalDecomposition>
    </AudioVisual>
    </MultimediaContent>
    </Description>
</Mpeg7>

```

11.4 Visual segment description tools

11.4.1 Introduction

This subclause specifies tools for describing segments of visual content such as images and video and their structural decompositions. The following table summarizes the tools specified in this subclause. Figure 35 shows the organization of the tools for describing segments of visual content.

Table 19 — Overview of the visual segment description tools

Tool	Functionality
StillRegion DS	This tool describes an image or a 2D spatial region of an image or a video frame. This tool extends the Segment DS.
Still region decomposition tools	These tools describe decompositions of still regions in space. These tools extend the base segment decomposition tools. The following still region decomposition tools are defined: StillRegionSpatialDecomposition DS: describes a spatial decomposition of a still region.
ImageText DS	This tool describes visual text in an image or a video frame. The ImageText DS extends the StillRegion DS.
Mosaic DS	This tool describes a mosaic or panoramic view of a video segment. The Mosaic DS extends the StillRegion DS.

<i>Tool</i>	<i>Functionality</i>
StillRegion3D DS	This tool describes a 3D image or a 3D spatial region of a 3D image. This tool extends the Segment DS.
3D still region decomposition tools	These tools describe decompositions of 3D still regions in space. These tools extend the base segment decomposition tools. The following 3D still region decomposition tools are defined: • StillRegion3DSpatialDecomposition DS: describes a spatial decomposition of a 3D still region.
VideoSegment DS	This tool describes a video or a temporal segment of a video. The VideoSegment DS extends the Segment DS.
Video segment decomposition tools	These tools describe decompositions of video segments in space, time, and media source. These tools extend the base segment decomposition tools. The following video segment decomposition tools are defined: • VideoSegmentSpatialDecomposition DS: describes a spatial decomposition of a video segment. • VideoSegmentTemporalDecomposition DS: describes a temporal decomposition of a video segment. • VideoSegmentSpatioTemporalDecomposition DS: describes a spatio-temporal decomposition of a video segment. • VideoSegmentMediaSourceDecomposition DS: describes a media source decomposition of a video segment.
MovingRegion DS	This tool describes a video or a spatio-temporal region of a video. The MovingRegion DS extends the Segment DS.
Moving region decomposition tools	These tools describe decompositions of moving regions in space, time, and media source. These tools extend the base segment decomposition tools. The following moving region decomposition tools are defined: • MovingRegionSpatialDecomposition DS: describes a spatial decomposition of a moving region. • MovingRegionTemporalDecomposition DS: describes a temporal decomposition of a moving region. • MovingRegionSpatioTemporalDecomposition DS: describes a spatio-temporal decomposition of a moving region. • MovingRegionMediaSourceDecomposition DS: describes a media source decomposition of a moving region.
VideoText DS	This tool describes visual text in video. The VideoText DS extends the MovingRegion DS.

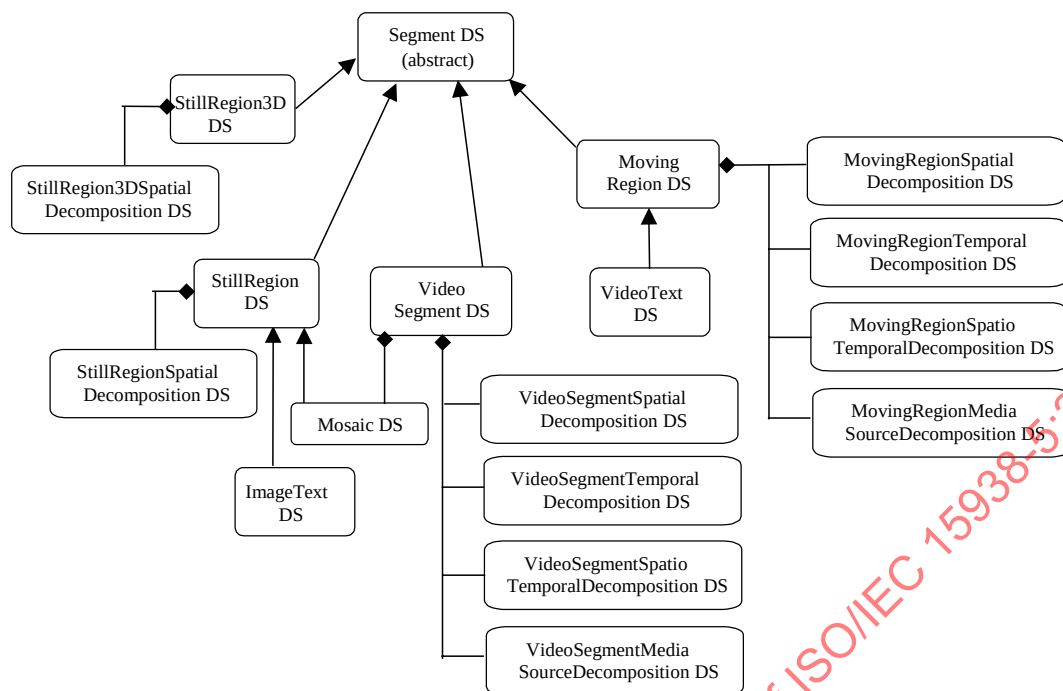


Figure 35 — Overview of the tools for describing segments of visual content

11.4.2 StillRegion DS

11.4.2.1 Introduction

The StillRegion DS describes an image or a 2D spatial region of an image or a video frame. The StillRegion DS extends the Segment DS. In the case of digital images, a still region can correspond to a single pixel, an arbitrary group of pixels, or the full image. The still region need not be connected in space. The SpatialMask D optionally describes a 2-D spatial mask that describes a set of non-connected sub-regions in the case that the still region is formed from non-connected sub-regions.

11.4.2.2 StillRegion DS syntax

```

<!-- ##### -->
<!-- Definition of StillRegion DS (11.4.2) -->
<!-- ##### -->

<!-- Definition of StillRegion DS -->
<complexType name="StillRegionType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <choice minOccurs="0">
          <element name="SpatialLocator" type="mpeg7:RegionLocatorType"/>
          <element name="SpatialMask" type="mpeg7:SpatialMaskType"/>
        </choice>
        <choice minOccurs="0">
          <element name="MediaTimePoint" type="mpeg7:mediaTimePointType"/>
          <element name="MediaRelTimePoint"
            type="mpeg7:MediaRelTimePointType"/>
          <element name="MediaRelIncrTimePoint"
            type="mpeg7:MediaRelIncrTimePointType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

<choice minOccurs="0" maxOccurs="unbounded">
  <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
  <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
  <element name="GridLayoutDescriptors" type="mpeg7:GridLayoutType"/>
</choice>
<element name="MultipleView" type="mpeg7:MultipleViewType"
  minOccurs="0"/>
<element name="SpatialDecomposition"
  type="mpeg7:StillRegionSpatialDecompositionType"
  minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</extension>
</complexContent>
</complexType>

```

11.4.2.3 StillRegion DS semantics

Semantics of the StillRegionType:

Name	Definition
StillRegionType	Describes an image or a 2D spatial region of an image or a video frame. The still region need not be connected in space. The StillRegion DS uses visual description tools (VisualDType and VisualDSType) to describe the visual features of the still region. The spatial localization or composition of the still region is optionally described using a choice of SpatialLocator or SpatialMask. StillRegionType extends SegmentType.
SpatialLocator	Describes the spatial localization of the still region using a RegionLocatorType (defined in ISO/IEC 15938-3) (optional). The spatial locator describes a spatially connected sub-region that includes the spatial components of the still region.
SpatialMask	Describes a 2-D spatial mask that defines the spatial composition of the still region (optional). The still region is formed from the set of sub-regions described by the SpatialMask. If absent, the still region is composed of the single connected region defined by the SpatialLocator.
MediaTimePoint	Indicates the time point of a still region from video using mediaTimePointType (optional).
MediaRelTimePoint	Indicates the time point of a still region from video using MediaRelTimePointType (optional).
MediaRelIncrTimePoint	Indicates the time point of a still region from video using MediaRelIncrTimePointType (optional).
VisualDescriptor	Describes visual features of the still region using a visual descriptor (optional). The specific VisualDTypes are defined in ISO/IEC 15938-3.
VisualDescriptionScheme	Describes visual features of the still region using a visual DS (optional). The specific VisualDSTypes are defined in ISO/IEC 15938-3.
GridLayoutDescriptors	Describes visual features of the sub-regions resulting from a grid decomposition of the still region (optional). GridLayoutDescriptors descriptions only apply to rectangular still regions. The GridLayoutType is defined in ISO/IEC

Name	Definition
	15938-3.
MultipleView	Describes visual features of a 3D object depicted in the 2D still region as seen from one or more viewing positions or angles (optional). The MultipleViewType is defined in ISO/IEC 15938-3.
SpatialDecomposition	Describes a spatial decomposition of the still region into one or more sub-segments (optional).

11.4.2.4 StillRegion DS examples (informative)

The following example illustrates the use of the StillRegion DS for describing the image shown in Figure 36. Various attributes of the image such as the location, annotation, and color are described using the following tools, respectively: MediaLocator datatype, the TextAnnotation datatype, and the ScalableColor D (defined in ISO/IEC 15938-3).

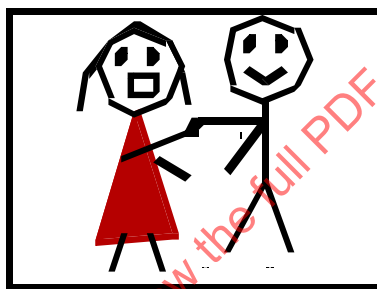


Figure 36 — Example of an image showing two people shaking hands. The image is described using the StillRegion DS

```

<StillRegion>
  <MediaLocator>
    <MediaUri>image.jpg</MediaUri>
  </MediaLocator>
  <TextAnnotation>
    <FreeTextAnnotation> Alex shakes hands with Ana </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>
</StillRegion>

```


11.4.3 Still region decomposition tools

11.4.3.1 Introduction

The still region decomposition tools extend the base segment decomposition tools. The tools describe the decomposition of an image or still region into one or more sub-segments in space.

11.4.3.2 Still region decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of StillRegionSpatialDecomposition DS (11.4.3) -->
<!-- ##### -->

<!-- Definition of StillRegionSpatialDecomposition DS -->
<complexType name="StillRegionSpatialDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatialSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="StillRegion" type="mpeg7:StillRegionType"/>
        <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

11.4.3.3 Still region decomposition tools semantics

Semantics of the StillRegionSpatialDecompositionType:

Name	Definition
StillRegionSpatialDecompositionType	Describes a spatial decomposition of a still region into one or more still regions. StillRegionSpatialDecompositionType extends SpatialSegmentDecompositionType.
StillRegion	Describes a still region that results from the spatial decomposition (optional).
StillRegionRef	Describes a reference to the description of a still region that results from the spatial decomposition (optional).

11.4.3.4 Still region decomposition tools examples (informative)

The following example illustrates the use of the StillRegionSpatialDecomposition DS in the StillRegion DS for describing the spatial decomposition of the image in Figure 36 into two still regions shown in Figure 37. In this example, the still region with id = "AlexAnaSR" corresponds to the full image from which two still regions, id = "AlexSR" and id = "AnaSR" are generated by spatial segmentation. In this example, each still region segment, id = "AlexSR", and id = "AnaSR", correspond to a person in the image. In this example, the segment decomposition has gaps (gaps = true) but no overlaps (overlap = false). The two still regions are spatially connected. The TextAnnotation describes an annotation of each still region. The ScalableColor D (defined in ISO/IEC 15938-3) describes the color of each still region.

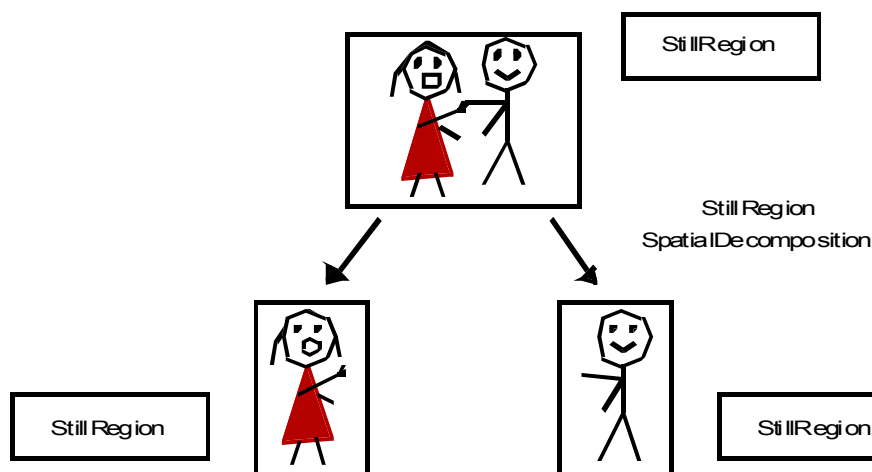


Figure 37 — Example of the spatial decomposition of an image into two still regions

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <SpatialDecomposition gap="true" overlap="false">
          <StillRegion id="AlexSR">
            <TextAnnotation>
              <FreeTextAnnotation> Alex </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>
          </StillRegion>
          <StillRegion id="AnaSR">
            <TextAnnotation>
              <FreeTextAnnotation> Ana </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>
          </StillRegion>
        </SpatialDecomposition>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

11.4.4 ImageText DS

11.4.4.1 Introduction

The ImageText DS describes a spatial region of an image or a video frame that visually contains text. The ImageText DS extends the StillRegion DS. The ImageText DS describes the textual characters, language, font size, and font style of the image text.

11.4.4.2 ImageText DS syntax

```

<!-- ##### -->
<!-- Definition of ImageText DS (11.4.4) -->
<!-- ##### -->

<!-- Definition of ImageText DS -->
<complexType name="ImageTextType">
  <complexContent>
    <extension base="mpeg7:StillRegionType">
      <sequence>
        <element name="Text" type="mpeg7:TextualType" minOccurs="0"/>
      </sequence>
      <attribute name="textType" use="optional">
        <simpleType>
          <union>
            <simpleType>
              <restriction base="NMTOKEN">
                <enumeration value="superimposed"/>
                <enumeration value="scene"/>
              </restriction>
            </simpleType>
            <simpleType>
              <restriction base="mpeg7:termReferenceType"/>
            </simpleType>
          </union>
        </simpleType>
      </attribute>
      <attribute name="fontSize" type="positiveInteger" use="optional"/>
      <attribute name="fontType" type="string" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

11.4.4.3 ImageText DS semantics

Semantics of the ImageTextType:

Name	Definition
ImageTextType	Describes a spatial region of an image or a video frame that visually contains text. ImageTextType extends the StillRegionType.
Text	Indicates the text string and the language of the image text (optional).
textType	Indicates the type of the image text (optional). The types are defined as follows: <i>superimposed</i> – the text is artificially embedded in the image or video frame such as captions or graphics (i.e., produced by title machines or graphical font generators). <i>scene</i> – the text appears naturally in the scene such as writing on people's shirts, signs or billboards in the scene, street names and shop names.
fontSize	Indicates the font size of the image text (optional).
fontType	Indicates the font type of the image text (optional).

11.4.4.4 ImageText DS examples (informative)

The following example illustrates the use of the ImageText DS for describing the image text of the image shown in Figure 38. The image contains one instance of scene image text ("Cherie, je suis la") and two instances of superimposed image text ("En direct du stade DE-KUIP Rotterdam Jean-Francois DEVELEY" and "Nous allons assister a un match fantastique").



Figure 38 — Example of an image that contains image texts

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <SpatialDecomposition gap="true" overlap="false">
          <StillRegion xsi:type="ImageTextType" textType="scene">
            <SpatialLocator>
              <!-- more elements here -->
            </SpatialLocator>
            <Text xml:lang="fr"> Cherie, je suis la </Text>
          </StillRegion>
          <StillRegion xsi:type="ImageTextType" textType="superimposed"
            fontSize="40" fontType="Arial">
            <SpatialLocator>
              <!-- more elements here -->
            </SpatialLocator>
            <Text xml:lang="fr">
              En direct du state DE-KUIP Rotterdam Jean-Francois DEVELEY </Text>
            </StillRegion>
            <StillRegion xsi:type="ImageTextType" textType="superimposed"
              fontSize="30" fontType="Times New Roman">
              <SpatialLocator>
                <!-- more elements here -->
              </SpatialLocator>
              <Text xml:lang="fr">
                Nous allons assister a un match fantastique </Text>
              </StillRegion>
            </SpatialDecomposition>
          </Image>
        </MultimediaContent>
      </Description>
    </Mpeg7>
```

11.4.5 Mosaic DS

11.4.5.1 Introduction

The Mosaic DS describes a mosaic or panoramic view of a video segment. The Mosaic DS extends the StillRegion DS. A mosaic can be constructed by aligning together and warping the frames of a video segment upon each other using a common spatial reference system. This gives a panoramic view of the scene as a single image. The Mosaic DS describes the motion warping parameters of the mosaic. A mosaic cannot be the result of any segment decomposition.

11.4.5.2 Mosaic DS syntax

```

<!-- ##### -->
<!-- Definition of Mosaic DS (11.4.5) -->
<!-- ##### -->

<!-- Definition of Mosaic DS -->
<complexType name="MosaicType">
  <complexContent>
    <extension base="mpeg7:StillRegionType">
      <sequence>
        <element name="WarpingParam" minOccurs="0">
          <complexType>
            <sequence minOccurs="1" maxOccurs="unbounded">
              <choice>
                <element name="MediaTimePoint"
                  type="mpeg7:mediaTimePointType"/>
                <element name="MediaRelTimePoint"
                  type="mpeg7:MediaRelTimePointType"/>
                <element name="MediaRelIncrTimePoint"
                  type="mpeg7:MediaRelIncrTimePointType"/>
              </choice>
              <element name="MotionParam" type="mpeg7:floatVector"/>
            </sequence>
            <attribute name="modelType" use="required">
              <simpleType>
                <union>
                  <simpleType>
                    <restriction base="NMTOKEN">
                      <enumeration value="translational"/>
                      <enumeration value="rotationAndScaling"/>
                      <enumeration value="affine"/>
                      <enumeration value="perspective"/>
                      <enumeration value="quadratic"/>
                    </restriction>
                  </simpleType>
                  <simpleType>
                    <restriction base="mpeg7:termReferenceType"/>
                  </simpleType>
                </union>
              </simpleType>
            </attribute>
            <attribute name="sourceSequenceWidth" type="positiveInteger"
              use="required"/>
            <attribute name="sourceSequenceHeight" type="positiveInteger"
              use="required"/>
            <attribute name="xOffset" type="float" use="required"/>
            <attribute name="yOffset" type="float" use="required"/>
            <attribute name="xOrigin" type="float" use="required"/>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        <attribute name="yOrigin" type="float" use="required"/>
    </complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

```

11.4.5.3 Mosaic DS semantics

Semantics of the MosaicType:

Name	Definition
MosaicType	Describes a mosaic of a video segment. A mosaic cannot be the result of any segment decomposition. MosaicType extends StillRegionType.
WarpingParam	Describes the warping parameters and reference information used to construct the mosaic (optional).
MediaTimePoint	Indicates the time point (e.g., frame number) for the motion parameters (MotionParam) relative to the first frame.
MediaRelTimePoint	Indicates the relative time point (e.g., frame number) for the motion parameters (MotionParam) relative to the first frame
MediaRelIncrTimePoint	Indicates the relative incremental time point (e.g., frame number) for the motion parameters (MotionParam) relative to the first frame
MotionParam	Indicates the motion parameters as a vector (see 11.4.5.3.1). The parameters of the model are given by the ParametricMotion D (the ParametricMotion D is defined in ISO/IEC 15938-3).
modelType	Indicates the type of the applied motion model. The types of models are given as follows: • <i>translational</i> – translational model. • <i>rotationAndScaling</i> – rotation and scaling model. • <i>affine</i> – affine model. • <i>perspective</i> – perspective model. • <i>quadratic</i> – quadratic motion model.
sourceSequenceWidth	Indicates the X dimension of the constructing video segment in pixel units.
sourceSequenceHeight	Indicates the Y dimension of the constructing video segment in pixel units.
xOffset	Indicates the X (pixel-) coordinate of the top-left pixel of the first frame of the video segment in the mosaic.
yOffset	Indicates the Y (pixel-) coordinate of the top-left pixel of the first frame of the video segment in the mosaic.
xOrigin	Indicates the X (pixel-) coordinate of the origin of the spatial reference coordinate system in the first frame.

Name	Definition
yOrigin	Indicates the Y (pixel-) coordinate of the origin of the spatial reference coordinate system in the first frame.

11.4.5.3.1 Mosaic warping parameters

The WarpingParam describes the connection between the mosaic and the video segment from which it is constructed. It contains sufficient information for reconstructing a video segment from the mosaic. The modelType specifies the type of the applied motion model. The types of motion models are defined in ISO/IEC 15938-3 as part of the ParametricMotion D:

- Translational model:
$$vx(x, y) = a_1, vy(x, y) = a_2$$
- Rotation/Scaling model:
$$vx(x, y) = a_1 + a_3 x + a_4 y, vy(x, y) = a_2 - a_4 x + a_3 y$$
- Affine model:
$$vx(x, y) = a_1 + a_3 x + a_4 y, vy(x, y) = a_2 + a_5 x + a_6 y$$
- Planar perspective model:
$$vx(x, y) = (a_1 + a_3 x + a_4 y) / (1 + a_7 x + a_8 y),$$
$$vy(x, y) = (a_2 + a_5 x + a_6 y) / (1 + a_7 x + a_8 y)$$
- Quadratic model:
$$vx(x, y) = a_1 + a_3 x + a_4 y + a_7 xy + a_9 x^2 + a_{10} y^2$$
$$vy(x, y) = a_2 + a_5 x + a_6 y + a_8 xy + a_{11} x^2 + a_{12} y^2$$

where vx(x, y) and vy(x, y) represent the x and y displacement components of the pixel of coordinates (x, y).

The fixed spatial reference system is defined in terms of the relation between the mosaic and the first frame of the video sequence. All values are specified in 'pixel' units. As illustrated in Figure 39, the xOffset and yOffset attributes specify the pixel coordinates of the top left pixel of the first frame in the mosaic. The origin of the coordinate system the motion parameters refer to is further shifted as specified by xorigin and yorigin attributes. This enables flexibility for positioning the spatial reference system. Finally, the dimensions of the video sequence are specified by sourceSequenceWidth and sourceSequenceHeight attributes, which allow the reconstruction of a corresponding output sequence.

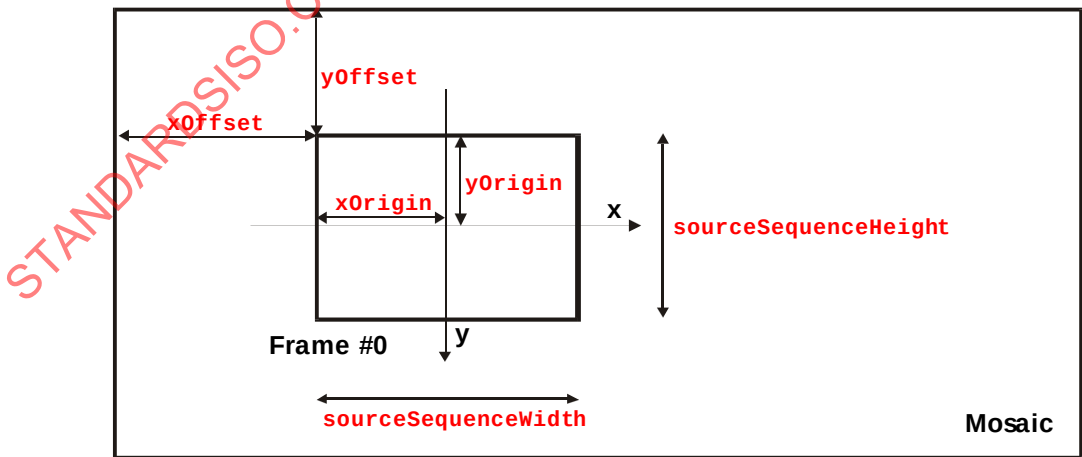


Figure 39 — Illustration of the spatial reference system in terms of the relation of the mosaic and the first frame of the video segment

A mosaic captures image information over a period of time. Therefore, temporal reference information shall be provided. In the case that a mosaic description is contained within a VideoSegment DS description, the time point and duration of the video segment is provided by the VideoSegment DS description. As illustrated in Figure 40, the vectors of motion parameters (MotionParam) are spread over this period. Each vector of motion parameters carries own temporal information in MediaTimePoint, MediaRelTimePoint, or MediaRelIncrTimePoint, which specifies the frame to which it is related. This provides flexibility for temporal spacing of the frames used for mosaicing. For instance, frame samples can be used such as every frame, every fifth frame, or any irregular spacing of frames.

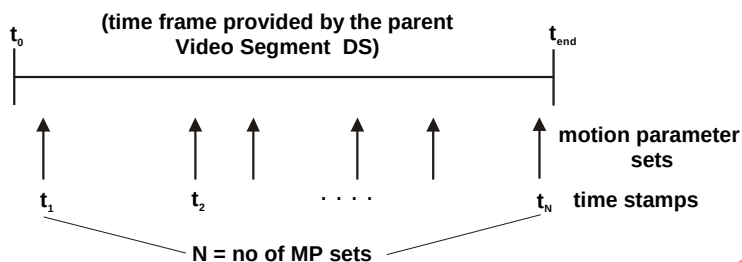


Figure 40 — Illustration of the temporal reference system for a mosaic defined in relation to the first frame of the video segment

11.4.5.4 Mosaic DS examples (informative)

The following example illustrates the use of the Mosaic DS for describing the mosaic of a video segment.

```
<Mosaic>
  <!-- Affine model, Center of Source -->
  <WarpingParam modelType="affine"
    sourceSequenceWidth="352"
    sourceSequenceHeight="288"
    xOffset="2" yOffset="7"
    xOrigin="176" yOrigin="144">
    <MediaTimePoint> <!-- more elements here --> </MediaTimePoint>
    <MotionParam> 0.0 0.0 0.0 0.0 0.0 0.0 </MotionParam>
    <MediaTimePoint> <!-- more elements here --> </MediaTimePoint>
    <MotionParam> -3.60 0.73 0.03 0.01 0.01 0.51 </MotionParam>
    <!-- more elements here -->
  </WarpingParam>
</Mosaic>
```

A mosaic allows the representation and visualization of a video segment using a single image. This functionality can also be provided by key frames. However, a mosaic contains more visual information than a key frame (e.g., motion warping parameters), but is not suitable for all kinds of shots (e.g., shots with rapidly changing background). Thus, mosaics and key frames serve complementary functions. Mosaics can also be useful for performing searching of video segments. Image features can be extracted from the mosaic image to describe the features of the video segment. Moreover, since the Mosaic DS extends the StillRegion DS, the StillRegionSpatialDecomposition DS can be used to describe the decomposition of a mosaic into still regions, each of which can be described individually. The warping information can also be used for summarizing the video segment since it describes the global motion in the segment. The mosaics may also be useful for video editing since the mosaics can be easily manipulated, using common image processing tools, and then reconstructing the video segment from the manipulated mosaic image.

11.4.6 StillRegion3D DS

11.4.6.1 Introduction

The StillRegion3D DS describes a 3D image or a spatial region of a 3D image. The StillRegion3D DS extends the Segment DS.

11.4.6.2 StillRegion3D DS syntax

```
<!-- ##### -->
<!-- Definition of StillRegion3D DS (11.4.6) -->
<!-- ##### -->

<!-- Definition of StillRegion3D DS -->
<complexType name="StillRegion3DType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
          <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
        </choice>
        <element name="MultipleView" type="mpeg7:MultipleViewType"
          minOccurs="0"/>
        <element name="SpatialDecomposition"
          type="mpeg7:StillRegion3DSpatialDecompositionType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.4.6.3 StillRegion3D DS semantics

Semantics of the StillRegion3DType:

Name	Definition
StillRegion3DType	Describes a 3D image or a 3D spatial region of a 3D image. StillRegion3DType extends SegmentType.
VisualDescriptor	Describes visual features of the 3-D still region using a visual descriptor (optional). The specific VisualDTypes are defined in ISO/IEC 15938-3.
VisualDescriptionScheme	Describes visual features of the 3D still region using a visual description scheme (optional). The specific VisualDSTypes are defined in ISO/IEC 15938-3.
MultipleView	Describes visual features of a 3D object depicted in the 3D still region as seen from one or more viewing positions or angles (optional). The MultipleViewType is defined in ISO/IEC 15938-3.
SpatialDecomposition	Describes the spatial decomposition of the 3D still region into one or more sub-segments (optional).

11.4.7 3D still region decomposition tools

11.4.7.1 Introduction

The 3D still region decomposition tools describe the decomposition of a 3D still region into one or more sub-segments in space. The 3D still region decomposition tools extend the base segment decomposition tools.

11.4.7.2 3D still region decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of StillRegion3DSpatialDecomposition DS (11.4.7) -->
<!-- ##### -->

<!-- Definition of StillRegion3DSpatialDecomposition DS -->
<complexType name="StillRegion3DSpatialDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatialSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="StillRegion" type="mpeg7:StillRegionType"/>
        <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
        <element name="StillRegion3D" type="mpeg7:StillRegion3DType"/>
        <element name="StillRegion3DRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

11.4.7.3 3D still region decomposition tools semantics

Semantics of the StillRegion3DSpatialDecompositionType:

Name	Definition
StillRegion3DSpatialDecompositionType	Describes the spatial decomposition of a 3D still region into one or more still regions and 3D still regions. StillRegion3DSpatialDecompositionType extends SpatialSegmentDecompositionType.
StillRegion	Describes a still region that results from the spatial decomposition (optional).
StillRegionRef	Describes a reference to the description of a still region that results from the spatial decomposition (optional).
StillRegion3D	Describes a 3D still region that results from the spatial decomposition (optional).
StillRegion3DRef	Describes a reference to the description of a 3D still region that results from the spatial decomposition (optional).

11.4.8 VideoSegment DS

11.4.8.1 Introduction

The VideoSegment DS describes a video or temporal segment of video. The VideoSegment DS extends the Segment DS. In the case of digital video, a video segment can correspond to a single frame, an arbitrary group of frames, or the full video sequence. The video segment need not be connected in time. The TemporalMask D optionally describes a temporal mask giving a set of non-connected sub-intervals in the case that the video segment is formed from non-connected sub-intervals.

11.4.8.2 VideoSegment DS syntax

```
<!-- ##### -->
<!-- Definition of VideoSegment DS (11.4.8) -->
<!-- ##### -->

<!-- Definition of VideoSegment DS -->
<complexType name="VideoSegmentType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <choice minOccurs="0">
          <element name="MediaTime" type="mpeg7:MediaTimeType"/>
          <element name="TemporalMask" type="mpeg7:TemporalMaskType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
          <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
          <element name="VisualTimeSeriesDescriptor"
            type="mpeg7:VisualTimeSeriesType"/>
        </choice>
        <element name="MultipleView" type="mpeg7:MultipleViewType"
          minOccurs="0"/>
        <element name="Mosaic" type="mpeg7:MosaicType"
          minOccurs="0" maxOccurs="unbounded"/>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SpatialDecomposition"
            type="mpeg7:VideoSegmentSpatialDecompositionType"/>
          <element name="TemporalDecomposition"
            type="mpeg7:VideoSegmentTemporalDecompositionType"/>
          <element name="SpatioTemporalDecomposition"
            type="mpeg7:VideoSegmentSpatioTemporalDecompositionType"/>
          <element name="MediaSourceDecomposition"
            type="mpeg7:VideoSegmentMediaSourceDecompositionType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.4.8.3 VideoSegment DS semantics

Semantics of the VideoSegmentType:

Name	Definition
VideoSegmentType	Describes a video or a temporal segment of a video. The VideoSegment DS uses visual description tools (VisualDType and VisualDSType) to describe the visual features of the video segment. The temporal localization or composition of the video segment is optionally described using a choice of MediaTime or TemporalMask. VideoSegmentType extends SegmentType.
MediaTime	Describes the temporal localization of the video segment using MediaTimeType (optional). The temporal locator describes a temporally connected sub-interval by specifying the start time and the duration of the video segment. The sub-interval describes a temporal bounding box that may bound a set of individual video segments. If neither MediaTime nor TemporalMask is described, then the video segment refers to the entire video.
TemporalMask	Describes a temporal mask that defines the temporal composition of the video segment. (optional). The video segment is formed from the set of sub-intervals described by the TemporalMask. If absent, the video segment is composed of the single connected interval defined by MediaTime.
VisualDescriptor	Describes visual features of the video segment using a visual descriptor (optional). The specific VisualDTypes are defined in ISO/IEC 15938-3.
VisualDescriptionScheme	Describes visual features of the video segment using a visual description scheme (optional). The specific VisualDSTypes are defined in ISO/IEC 15938-3.
VisualTimeSeriesDescriptor	Describes a temporal sequence of visual features in the video segment (optional). The VisualTimeSeriesDescriptor applies only in the case of a connected video segment. The VisualTimeSeriesDescriptorType is defined in ISO/IEC 15938-3.
MultipleView	Describes visual features of a 3D moving physical object depicted in the video segment as seen from one or more viewing positions or angles (optional). The MultipleViewType is defined in ISO/IEC 15938-3.
Mosaic	Describes a mosaic constructed from the video segment (optional).
SpatialDecomposition	Describes a spatial decomposition of the video segment into one or more sub-segments (optional).
TemporalDecomposition	Describes a temporal decomposition of the video segment into one or more sub-segments (optional).
SpatioTemporalDecomposition	Describes a spatio-temporal decomposition of the video segment into one or more sub-segments (optional).
MediaSourceDecomposition	Describes a media source decomposition of the video segment into one or more sub-segments (optional).

11.4.8.4 VideoSegment DS examples (informative)

The following example illustrates the use of the VideoSegment DS for describing the video shown in Figure 25. In this example, the video segment id = "RootVS" represents the full video. The MediaTime describes the time information for the segment. The GoFGoPColor D (defined in ISO/IEC 15938-3) describes the color information of the video segment.

```
<VideoSegment id="RootVS">
  <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>
</VideoSegment>
```

11.4.9 Video segment decomposition tools

11.4.9.1 Introduction

The video segment decomposition tools extend the base segment decomposition tools. The video segment decomposition tools describe the decomposition of a video segment into one or more sub-segments in space, time, and media source.

11.4.9.2 Video segment decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of VideoSegment decomposition DSs (11.4.9) -->
<!-- ##### -->

<!-- Definition of VideoSegmentSpatialDecomposition DS -->
<complexType name="VideoSegmentSpatialDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatialSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
        <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of VideoSegmentTemporalDecomposition DS -->
<complexType name="VideoSegmentTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:TemporalSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="VideoSegment" type="mpeg7:VideoSegmentType"/>
        <element name="VideoSegmentRef" type="mpeg7:ReferenceType"/>
        <element name="StillRegion" type="mpeg7:StillRegionType"/>
        <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

```

<!-- Definition of VideoSegmentSpatioTemporalDecomposition DS -->
<complexType name="VideoSegmentSpatioTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatioTemporalSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
        <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
        <element name="StillRegion" type="mpeg7:StillRegionType"/>
        <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of VideoSegmentMediaSourceDecomposition DS -->
<complexType name="VideoSegmentMediaSourceDecompositionType">
  <complexContent>
    <extension base="mpeg7:MediaSourceSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="VideoSegment" type="mpeg7:VideoSegmentType"/>
        <element name="VideoSegmentRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

11.4.9.3 Video segment decomposition tools semantics

Semantics of the VideoSegmentSpatialDecompositionType:

Name	Definition
VideoSegmentSpatialDecompositionType	Describes the spatial decomposition of a video segment into one or more moving regions. VideoSegmentSpatialDecompositionType extends SpatialSegmentDecompositionType.
MovingRegion	Describes a moving region that results from the spatial decomposition (optional).
MovingRegionRef	Describes a reference to the description of a moving region that results from the spatial decomposition (optional).

Semantics of the VideoSegmentTemporalDecompositionType:

Name	Definition
VideoSegmentTemporalDecompositionType	Describes the temporal decomposition of a video segment into one or more video segments and still regions. VideoSegmentTemporalDecompositionType extends TemporalSegmentDecompositionType.
VideoSegment	Describes a video segment that results from the temporal decomposition (optional).

<i>Name</i>	<i>Definition</i>
VideoSegmentRef	Describes a reference to the description of a video segment that results from the temporal decomposition (optional).
StillRegion	Describes a still region that results from the temporal decomposition (optional).
StillRegionRef	Describes a reference to the description of a still region that results from the temporal decomposition (optional).

Semantics of the VideoSegmentSpatioTemporalDecompositionType:

<i>Name</i>	<i>Definition</i>
VideoSegmentSpatioTemporalDecompositionType	Describes the spatio-temporal decomposition of a video segment into one or more moving regions and still regions. VideoSegmentSpatioTemporalDecompositionType extends SpatioTemporalSegmentDecompositionType.
MovingRegion	Describes a moving region that results from the spatio-temporal decomposition (optional).
MovingRegionRef	Describes a reference to the description of a moving region that results from the spatio-temporal decomposition (optional).
StillRegion	Describes a still region that results from the spatio-temporal decomposition (optional).
StillRegionRef	Describes a reference to the description of a still region that results from the spatio-temporal decomposition (optional).

Semantics of the VideoSegmentMediaSourceDecompositionType:

<i>Name</i>	<i>Definition</i>
VideoSegmentMediaSourceDecompositionType	Describes a media source decomposition of a video segment into one or more video segments. VideoSegmentMediaSourceDecompositionType extends MediaSourceSegmentDecompositionType.
VideoSegment	Describes a video segment that results from the media source decomposition (optional).
VideoSegmentRef	Describes a reference to the description of a video segment that results from the media source decomposition (optional).

11.4.9.4 Video segment decomposition tools examples (informative)

The following examples illustrate the use of the VideoSegmentTemporalDecomposition DS and VideoSegmentSpatioTemporalDecomposition DS for describing the temporal and spatio-temporal decomposition of a video segment.

The first example describes the segment decomposition shown in Figure 25. In this example, the video segment has two scenes, "Chase" and "Capture". The video segment id = "RootVS" is decomposed in two video segments, id = "ChaseVS" and id = "CaptureVS". The video segment id = "ChaseVS" corresponds to the scene "Chase", and video segment id = "CaptureVS" corresponds to the scene "Capture". The segment decomposition has neither gaps (gap = false) nor overlaps (overlap = false). The video segments are temporally connected. At the same time, the video segment "CaptureVS" is decomposed into two moving regions: id = "ManMR" and id = "WomanMR". The moving region id = "ManMR" corresponds to the object "Man" and moving region id = "WomanMR" corresponds to the object "Woman". This segment decomposition has gaps and overlaps (gap = true and overlap = true). Each of the moving regions is connected. The GoFGoPColor D (defined in ISO/IEC 15938-3) describes the color information of the video segment.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="RootVS">
        <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>
        <TemporalDecomposition gap="false" overlap="false">
          <VideoSegment id="ChaseVS">
            <TextAnnotation>
              <FreeTextAnnotation> Chase </FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:00:00</MediaTimePoint>
              <MediaDuration>PT0M15S</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>
          </VideoSegment>
          <VideoSegment id="CaptureVS">
            <TextAnnotation>
              <FreeTextAnnotation> Capture </FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:00:15</MediaTimePoint>
              <MediaDuration>PT1M15S</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>
          </VideoSegment>
        </TemporalDecomposition>
      </Video>
    </MultimediaContent>
  </Description>
```



```

        <SpatioTemporalDecomposition gap="true" overlap="true">
          <MovingRegion id="ManMR">
            <TextAnnotation>
              <FreeTextAnnotation> Man </FreeTextAnnotation>
            </TextAnnotation>
          </MovingRegion>
          <MovingRegion id="WomanMR">
            <TextAnnotation>
              <FreeTextAnnotation> Woman </FreeTextAnnotation>
            </TextAnnotation>
          </MovingRegion>
        </SpatioTemporalDecomposition>
      </VideoSegment>
    </TemporalDecomposition>
  </Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

The second example describes the video segment decomposition shown in Figure 29 (b). In this example, video segment id = "VS1" represents a video consisting of four segments: " id = VS2", id = "VS3", id = "VS4", and id = "VS5". The segment decomposition has neither gaps nor overlaps (gap = false and overlap = false). The video segment id = "VS1" is not connected in time; it is composed of two temporal intervals of duration six and three minutes, respectively. Each video segment id = "VS2", id = "VS3", id = "VS4", and id = "VS5" is temporally connected.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="VS1">
        <TemporalMask>
          <SubInterval>
            <MediaTimePoint>T00:00:00</MediaTimePoint>
            <MediaDuration>PT6M</MediaDuration>
          </SubInterval>
          <SubInterval>
            <MediaTimePoint>T00:07:00</MediaTimePoint>
            <MediaDuration>PT3M</MediaDuration>
          </SubInterval>
        </TemporalMask>
        <TemporalDecomposition gap="false" overlap="false">
          <VideoSegment id="VS2">
            <MediaTime>
              <MediaTimePoint>T00:00:00</MediaTimePoint>
              <MediaDuration>PT5M</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>
          </VideoSegment>
          <VideoSegment id="VS3">
            <MediaTime>
              <MediaTimePoint>T00:05:00</MediaTimePoint>
              <MediaDuration>PT1M</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>
</VideoSegment>
<VideoSegment id="VS4">
    <MediaTime>
        <MediaTimePoint>T00:07:00</MediaTimePoint>
        <MediaDuration>PT2M</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>
</VideoSegment>
<VideoSegment id="VS5">
    <MediaTime>
        <MediaTimePoint>T00:09:00</MediaTimePoint>
        <MediaDuration>PT1M</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>
</VideoSegment>
</TemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

11.4.10 MovingRegion DS

11.4.10.1 Introduction

The MovingRegion DS describes a video or a spatio-temporal region of a video. The MovingRegion DS extends the Segment DS. In the case of digital video, a moving region can correspond to an arbitrary group of pixels, or the full video. The moving region need not be connected in space or time. The SpatioTemporalMask D optionally describes a spatio-temporal mask giving a set of non-connected sub-regions in the case that the moving region is formed from non-connected sub-regions.

11.4.10.2 MovingRegion DS syntax

```

<!-- ##### -->
<!-- Definition of MovingRegion DS (11.4.10) -->
<!-- ##### -->

<!-- Definition of MovingRegion DS -->
<complexType name="MovingRegionType">
    <complexContent>
        <extension base="mpeg7:SegmentType">
            <sequence>
                <choice minOccurs="0">

```

```

        <element name="SpatioTemporalLocator"
            type="mpeg7:SpatioTemporalLocatorType"/>
        <element name="SpatioTemporalMask"
            type="mpeg7:SpatioTemporalMaskType"/>
    </choice>
    <choice minOccurs="0" maxOccurs="unbounded">
        <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
        <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
        <element name="VisualTimeSeriesDescriptor"
            type="mpeg7:VisualTimeSeriesType"/>
    </choice>
    <element name="MultipleView" type="mpeg7:MultipleViewType"
        minOccurs="0"/>
    <choice minOccurs="0" maxOccurs="unbounded">
        <element name="SpatialDecomposition"
            type="mpeg7:MovingRegionSpatialDecompositionType"/>
        <element name="TemporalDecomposition"
            type="mpeg7:MovingRegionTemporalDecompositionType"/>
        <element name="SpatioTemporalDecomposition"
            type="mpeg7:MovingRegionSpatioTemporalDecompositionType"/>
        <element name="MediaSourceDecomposition"
            type="mpeg7:MovingRegionMediaSourceDecompositionType"/>
    </choice>
    </sequence>
</extension>
</complexContent>
</complexType>

```

11.4.10.3 MovingRegion DS semantics

Semantics of the MovingRegionType:

Name	Definition
MovingRegionType	Describes a video or a spatio-temporal region of a video. The moving region need not be connected in space or time. The MovingRegion DS uses visual description tools (VisualDType and VisualDSType) to describe the visual features of the moving region. The spatio-temporal localization or composition of the moving region is optionally described using a choice of SpatioTemporalLocator or SpatioTemporalMask. MovingRegionType extends SegmentType.
SpatioTemporalLocator	Describes the spatio-temporal localization of the moving region using SpatioTemporalLocatorType (optional). The spatio-temporal locator describes a spatio-temporally connected sub-region that includes the spatio-temporal components of the moving region. The SpatioTemporalLocatorType is defined in ISO/IEC 15938-3.
SpatioTemporalMask	Describes a spatio-temporal mask that defines the spatio-temporal composition of the moving region (optional). The moving region is formed from the set of sub-regions described by the SpatioTemporalMask. If absent, the moving region is composed of the single connected region defined by the SpatioTemporalLocator.
VisualDescriptor	Describes the visual features of the moving region using a visual descriptor (optional). The specific VisualDTypes are defined in ISO/IEC 15938-3.
VisualDescriptionScheme	Describes the visual features of the moving region using a visual description scheme (optional). The specific VisualDSTypes are defined

Name	Definition
	in ISO/IEC 15938-3.
VisualTimeSeriesDescriptor	Describes a temporal sequence of visual features in the moving region (optional). The VisualTimeSeriesDescriptor applies only in the case of a connected moving region. The VisualTimeSeriesType is defined in ISO/IEC 15938-3.
MultipleView	Describes visual features of a 3D moving physical object depicted in the moving region as seen from one or more viewing positions or angles (optional). The MultipleViewType is defined in ISO/IEC 15938-3.
SpatialDecomposition	Describes the spatial decomposition of the moving region into one or more sub-segments (optional).
TemporalDecomposition	Describes the temporal decomposition of the moving region into one or more sub-segments (optional).
SpatioTemporalDecomposition	Describes the spatio-temporal decomposition of the moving region into one or more sub-segments (optional).
MediaSourceDecomposition	Describes the media source decomposition of the moving region into one or more sub-segments (optional).

11.4.10.4 MovingRegion DS examples (informative)

The following example illustrates the use of the MovingRegion DS for describing the spatio-temporal region corresponding to the object "Man" in the video shown in Figure 25. In this example, the moving region id = "ManMR" corresponds to the object "Man".

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <SpatioTemporalDecomposition>
          <MovingRegion id="ManMR">
            <TextAnnotation>
              <FreeTextAnnotation> Man </FreeTextAnnotation>
            </TextAnnotation>
            <SpatioTemporalLocator>
              <FigureTrajectory type="rectangle">
                <MediaTime>
                  <MediaTimePoint>T00:00:15</MediaTimePoint>
                  <MediaDuration>PT1M15S</MediaDuration>
                </MediaTime>
                <Vertex>
                  <WholeInterval>
                    <MediaDuration>PT0S</MediaDuration>
                  </WholeInterval>
                  <InterpolationFunctions>
                    <KeyValue type="startPoint">0.1</KeyValue>
                    <KeyValue type="startPoint">0.2</KeyValue>
                  </InterpolationFunctions>
                </Vertex>
                <Vertex>
                  <WholeInterval>
                    <MediaDuration>PT1S</MediaDuration>

```

```

        </WholeInterval>
        <InterpolationFunctions>
            <KeyValue type="startPoint">0.3</KeyValue>
            <KeyValue type="startPoint">0.4</KeyValue>
        </InterpolationFunctions>
    </Vertex>
    <Vertex>
        <WholeInterval>
            <MediaDuration>PT2S</MediaDuration>
        </WholeInterval>
        <InterpolationFunctions>
            <KeyValue type="startPoint">0.5</KeyValue>
            <KeyValue type="startPoint">0.5</KeyValue>
        </InterpolationFunctions>
    </Vertex>
    <!-- more elements here -->
</FigureTrajectory>
</SpatioTemporalLocator>
</MovingRegion>
</SpatioTemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

11.4.11 Moving region decomposition tools

11.4.11.1 Introduction

The moving region decomposition tools describe the decomposition of a moving region into one or more sub-segments in space, time, and media source. The moving region decomposition tools extend the base segment decomposition tools.

11.4.11.2 Moving region decomposition tools syntax

```

<!-- ##### -->
<!-- Definition of MovingRegion decomposition DSs (11.4.11) -->
<!-- ##### -->

<!-- Definition of MovingRegionSpatialDecomposition DS -->
<complexType name="MovingRegionSpatialDecompositionType">
    <complexContent>
        <extension base="mpeg7:SpatialSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
                <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of MovingRegionTemporalDecomposition DS -->
<complexType name="MovingRegionTemporalDecompositionType">
    <complexContent>
        <extension base="mpeg7:TemporalSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
                <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

```

```

        <element name="StillRegion" type="mpeg7:StillRegionType"/>
        <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
    </choice>
</extension>
</complexContent>
</complexType>

<!-- Definition of MovingRegionSpatioTemporalDecomposition DS -->
<complexType name="MovingRegionSpatioTemporalDecompositionType">
    <complexContent>
        <extension base="mpeg7:SpatioTemporalSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
                <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
                <element name="StillRegion" type="mpeg7:StillRegionType"/>
                <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of MovingRegionMediaSourceDecomposition DS -->
<complexType name="MovingRegionMediaSourceDecompositionType">
    <complexContent>
        <extension base="mpeg7:MediaSourceSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="MovingRegion" type="mpeg7:MovingRegionType"/>
                <element name="MovingRegionRef" type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

```

11.4.11.3 Moving region decomposition tools semantics

Semantics of the MovingRegionSpatialDecompositionType:

Name	Definition
MovingRegionSpatialDecompositionType	Describes the spatial decomposition of a moving region into one or more moving regions. MovingRegionSpatialDecompositionType extends SpatialSegmentDecompositionType.
MovingRegion	Describes a moving region that results from the spatial decomposition (optional).
MovingRegionRef	Describes a reference to the description of a moving region that results from the spatial decomposition (optional).

Semantics of the MovingRegionTemporalDecompositionType:

<i>Name</i>	<i>Definition</i>
MovingRegionTemporalDecompositionType	Describes the temporal decomposition of a moving region into one or more moving regions and still regions. MovingRegionTemporalDecompositionType extends TemporalSegmentDecompositionType.
MovingRegion	Describes a moving region that results from the temporal decomposition (optional).
MovingRegionRef	Describes a reference to the description of a moving region that results from the temporal decomposition (optional).
StillRegion	Describes a still region that results from the temporal decomposition (optional).
StillRegionRef	Describes a reference to the description of a still region that results from the temporal decomposition (optional).

Semantics of the MovingRegionSpatioTemporalDecompositionType:

<i>Name</i>	<i>Definition</i>
MovingRegionSpatioTemporalDecompositionType	Describes the spatio-temporal decomposition of a video segment into one or more moving regions and still regions. MovingRegionSpatioTemporalDecompositionType extends SpatioTemporalSegmentDecompositionType.
MovingRegion	Describes a moving region that results from the spatio-temporal decomposition (optional).
MovingRegionRef	Describes a reference to the description of a moving region that results from the spatio-temporal decomposition (optional).
StillRegion	Describes a still region that results from the spatio-temporal decomposition (optional).
StillRegionRef	Describes a reference to the description of a still region that results from the spatio-temporal decomposition (optional).

Semantics of the MovingRegionMediaSourceDecompositionType:

<i>Name</i>	<i>Definition</i>
MovingRegionMediaSourceDecompositionType	Describes a media source decomposition of a moving region into one or more moving regions. MovingRegionMediaSourceDecompositionType extends MediaSourceSegmentDecompositionType.
MovingRegion	Describes a moving region that results from the media source decomposition (optional).
MovingRegionRef	Describes a reference to the description of a moving region that results from the media source decomposition (optional).

11.4.11.4 Moving region decomposition tools examples (informative)

The following example illustrates the use of the MovingRegionTemporalDecomposition DS for describing a video object plane or key still region of the moving region id = "ManMR" in 11.3.2.5.4. In this example, the still region id = "ManKeySR" represents the still region of the object "Man" in a video frame. The segment decomposition has gaps but no overlaps.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="VS1">
        <SpatioTemporalDecomposition>
          <MovingRegion id="ManMR">
            <TextAnnotation>
              <FreeTextAnnotation> Man (moving region)</FreeTextAnnotation>
            </TextAnnotation>
            <TemporalDecomposition gap="true" overlap="false">
              <StillRegion id="ManKeySR">
                <MediaLocator>
                  <MediaUri>image.jpg</MediaUri>
                </MediaLocator>
                <TextAnnotation>
                  <FreeTextAnnotation> Man (still region)
                </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>
            </StillRegion>
          </TemporalDecomposition>
        </MovingRegion>
      </SpatioTemporalDecomposition>
    </Video>
  </MultimediaContent>
</Description>
</Mpeg7>
```

11.4.12 VideoText DS

11.4.12.1 Introduction

The VideoText DS describes a spatio-temporal region of video that visually contains text. The VideoText DS extends the MovingRegion DS. The VideoText DS describes the textual characters, language, font size, and font style of the video text.

11.4.12.2 VideoText DS syntax

```
<!-- ##### -->
<!-- Definition of VideoText DS (11.4.12) -->
<!-- ##### -->

<!-- Definition of VideoText DS -->
<complexType name="VideoTextType">
  <complexContent>
    <extension base="mpeg7:MovingRegionType">
      <sequence>
        <element name="Text" type="mpeg7:TextualType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```



```
</sequence>
<attribute name="textType" use="optional">
  <simpleType>
    <union>
      <simpleType>
        <restriction base="NMTOKEN">
          <enumeration value="superimposed"/>
          <enumeration value="scene"/>
        </restriction>
      </simpleType>
      <simpleType>
        <restriction base="mpeg7:termReferenceType"/>
      </simpleType>
    </union>
  </simpleType>
</attribute>
<attribute name="fontSize" type="positiveInteger" use="optional"/>
<attribute name="fontType" type="string" use="optional"/>
</extension>
</complexContent>
</complexType>
```

11.4.12.3 VideoText DS semantics

Semantics of the VideoTextType:

Name	Definition
VideoTextType	Describes a spatio-temporal region of video that visually contains text. VideoTextType extends MovingRegionType.
Text	Indicates the text string and the language of the video text (optional).
textType	Indicates the type of the video text (optional). The types are defined as follows: - superimposed – the text is artificially embedded in the video such as captions or graphics (i.e., produced by title machines or graphical font generators). - scene – the text appears naturally in the scene such as writing on people's shirts, signs or billboards in the scene, street names and shop names.
fontSize	Indicates the font size of the video text (optional).
fontType	Indicates the font type of the video text (optional).

11.4.12.4 VideoText DS examples (informative)

The following example illustrates the use of the VideoText DS for describing a superimposed video text with text string "Victory".

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="VS1">
        <SpatioTemporalDecomposition>
          <MovingRegion xsi:type="VideoTextType" id="VideoText1">
```

```

        textType="superimposed" fontSize="40" fontType="Courier New">
        <Text> Victory </Text>
    </MovingRegion>
</SpatioTemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>

```

11.5 Audio segment description tools

11.5.1 Introduction

This subclause specifies tools for describing segments of audio content and their structural decompositions. The following table summarizes the tools specified in this subclause. Figure 41 shows the organization of the tools for describing segments of audio content.

Table 20 — Overview of the audio segment description tools

Tool	Functionality
AudioSegment DS	This tool describes audio and segments of audio. This tool extends the Segment DS.
Audio segment decomposition tools	These tools describe decompositions of audio segments in time and media source. These tools extend the base segment decomposition tools. The following audio segment decomposition tools are defined: - AudioSegmentTemporalDecomposition DS: describes the temporal decomposition of an audio segment. - AudioSegmentMediaSourceDecomposition DS: describes the media source decomposition of an audio segment.

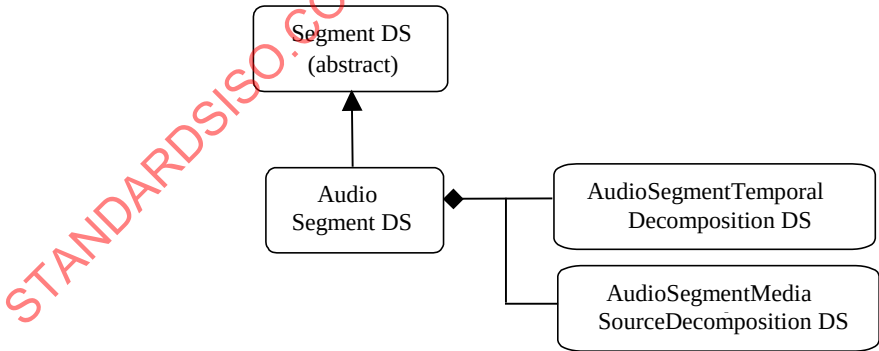


Figure 41 — Overview of the tools for describing segments of audio content

11.5.2 AudioSegment DS

11.5.2.1 Introduction

The AudioSegment DS describes audio or temporal segment of audio. The AudioSegment DS extends the Segment DS. In the case of digital audio, a audio segment can correspond to an arbitrary group of samples, or the full audio sequence. The audio segment need not be connected in time. The TemporalMask D optionally describes a temporal mask giving a set of non-connected sub-intervals in the case that the audio segment is formed from non-connected sub-intervals.

11.5.2.2 AudioSegment DS syntax

```
<!-- ##### -->
<!-- Definition of AudioSegment DS (11.5.2) -->
<!-- ##### -->

<!-- Definition of AudioSegment DS -->
<complexType name="AudioSegmentType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <choice minOccurs="0">
          <element name="MediaTime" type="mpeg7:MediaTimeType"/>
          <element name="TemporalMask" type="mpeg7:TemporalMaskType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="AudioDescriptor" type="mpeg7:AudioDType"/>
          <element name="AudioDescriptionScheme" type="mpeg7:AudioDSType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="TemporalDecomposition"
            type="mpeg7:AudioSegmentTemporalDecompositionType"/>
          <element name="MediaSourceDecomposition"
            type="mpeg7:AudioSegmentMediaSourceDecompositionType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.5.2.3 AudioSegment DS semantics

Semantics of the AudioSegmentType:

Name	Definition
AudioSegmentType	Describes audio or a temporal segment of audio. The AudioSegment DS uses audio description tools (AudioDType and AudioDSType) to describe the audio features of the audio segment. The temporal localization or composition of the audio segment is optionally described using a choice of MediaTime or TemporalMask. AudioSegmentType extends SegmentType.
MediaTime	Describes the temporal localization of the audio segment using MediaTimeType (optional). The temporal locator describes a temporally connected sub-interval by specifying the start time and the duration of the audio segment. The sub-interval describes a temporal bounding box that may bound a set of individual audio segments. If neither MediaTime nor

Name	Definition
	TemporalMask is described, then the audio segment refers to the entire audio.
TemporalMask	Describes a temporal mask that defines the temporal composition of the audio segment (optional). The audio segment is formed from the set of sub-intervals described by the TemporalMask. If absent, the audio segment is composed of the single connected interval defined by MediaTime.
AudioDescriptor	Describes audio features of the audio segment using an audio descriptor (optional). The specific AudioDTypes are defined in ISO/IEC 15938-4.
AudioDescriptionScheme	Describes audio features of the audio segment using an audio description scheme (optional). The specific AudioDSTypes are defined in ISO/IEC 15938-4.
TemporalDecomposition	Describes the temporal decomposition of the audio segment into one or more sub-segments (optional).
MediaSourceDecomposition	Describes the media source decomposition of the audio segment into one or more sub-segments (optional).

11.5.2.4 AudioSegment DS examples (informative)

The following examples illustrate the use of the AudioSegment DS for describing audio content. The MediaTime describes the time information for the audio segment.

```
<AudioSegment>
  <MediaTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT1M30S</MediaDuration>
  </MediaTime>
</AudioSegment>
```

11.5.3 Audio segment decomposition tools

11.5.3.1 Introduction

The audio segment decomposition tools describe the decomposition of an audio segment into one or more sub-segments in time and media source. The audio segment decomposition tools extend the base segment decomposition tools.

11.5.3.2 Audio segment decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of AudioSegment decomposition DSs (11.5.3) -->
<!-- ##### -->

<!-- Definition of AudioSegmentTemporalDecomposition DS -->
<complexType name="AudioSegmentTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:TemporalSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="AudioSegment" type="mpeg7:AudioSegmentType"/>
        <element name="AudioSegmentRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

```

        </choice>
      </extension>
    </complexContent>
  </complexType>

  <!-- Definition of AudioSegmentMediaSourceDecomposition DS -->
  <complexType name="AudioSegmentMediaSourceDecompositionType">
    <complexContent>
      <extension base="mpeg7:MediaSourceSegmentDecompositionType">
        <choice minOccurs="1" maxOccurs="unbounded">
          <element name="AudioSegment" type="mpeg7:AudioSegmentType"/>
          <element name="AudioSegmentRef" type="mpeg7:ReferenceType"/>
        </choice>
      </extension>
    </complexContent>
  </complexType>

```

11.5.3.3 Audio segment decomposition tools semantics

Semantics of the AudioSegmentTemporalDecompositionType:

Name	Definition
AudioSegmentTemporalDecompositionType	Describes the temporal decomposition of an audio segment into one or more audio segments. AudioSegmentTemporalDecompositionType extends TemporalSegmentDecompositionType.
AudioSegment	Describes an audio segment that results from the temporal decomposition (optional).
AudioSegmentRef	Describes a reference to the descriptions of an audio segment that results from the temporal decomposition (optional).

Semantics of the AudioSegmentMediaSourceDecompositionType:

Name	Definition
AudioSegmentMediaSourceDecompositionType	Describes the media source decomposition of an audio segment into one or more audio segments. AudioSegmentMediaSourceDecompositionType extends MediaSourceSegmentDecompositionType
AudioSegment	Describes an audio segment that results from the media source decomposition (optional).
AudioSegmentRef	Describes a reference to the description of an audio segment that results from the media source decomposition (optional).

11.5.3.4 Audio segment decomposition tools examples (informative)

The following example illustrates the use of the AudioSegmentTemporalDecomposition DS for describing the temporal decomposition of audio content.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioType">
      <Audio>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT1M30S</MediaDuration>
        </MediaTime>
        <TemporalDecomposition gap="false" overlap="false">
          <AudioSegment>
            <TextAnnotation>
              <FreeTextAnnotation> Chase </FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:00:00</MediaTimePoint>
              <MediaDuration>PT0M15S</MediaDuration>
            </MediaTime>
          </AudioSegment>
          <AudioSegment>
            <TextAnnotation>
              <FreeTextAnnotation> Capture </FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:00:15</MediaTimePoint>
              <MediaDuration>PT1M15S</MediaDuration>
            </MediaTime>
          </AudioSegment>
        </TemporalDecomposition>
      </Audio>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

11.6 Audio-visual segment description tools

11.6.1 Introduction

This subclause specifies tools for describing segments of AV content and their structural decompositions. The following table summarizes the tools specified in this subclause. Figure 42 shows the organization of the tools for describing segments of AV content.

Table 21 — Overview of the AV segment description tools

<i>Tool</i>	<i>Functionality</i>
AudioVisualSegment DS	This tool describes AV content and segments of AV content. This tool extends the Segment DS.
Audio-visual segment decomposition tools	These tools describe decompositions of AV segments in space, time, and media source. These tools extend the base segment decomposition tools. The following AV segment decomposition tools are defined: AudioVisualSegmentSpatialDecomposition DS: describes the spatial decomposition of an AV segment.

Tool	Functionality
	- AudioVisualSegmentTemporalDecomposition DS: describes the temporal decomposition of an AV segment. - AudioVisualSegmentSpatioTemporalDecomposition DS: describes the spatio-temporal decomposition of an AV segment. - AudioVisualSegmentMediaSourceDecomposition DS: describes the media source decomposition of an AV segment.
AudioVisualRegion DS	This tool describes AV content and arbitrary spatio-temporal regions of AV content. This tool extends the Segment DS.
Audio-visual region decomposition tools	These tools describe decompositions of AV regions in space, time, and media source. These tools extend the base segment decomposition tools. The following AV region decomposition tools are defined: - AudioVisualRegionSpatialDecomposition DS: describes the spatial decomposition of an AV region. - AudioVisualRegionTemporalDecomposition DS: describes the temporal decomposition of an AV region. - AudioVisualRegionSpatioTemporalDecomposition DS: describes the spatio-temporal decomposition of an AV region. - AudioVisualRegionMediaSourceDecomposition DS: describes the media source decomposition of an AV region.

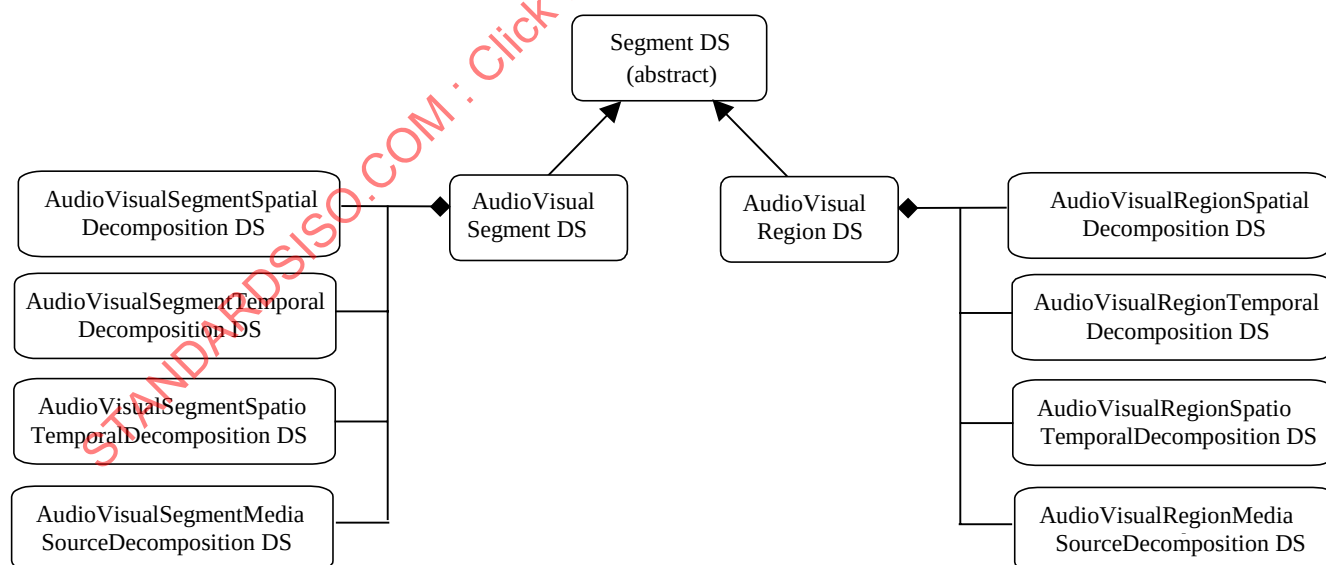


Figure 42 — Overview of the tools for describing segments of AV content

11.6.2 AudioVisualSegment DS

11.6.2.1 Introduction

The AudioVisualSegment DS describes AV content or temporal segment of AV content, which corresponds to the audio and video content in the same temporal intervals. The AudioVisualSegment DS extends the Segment DS. The AV segment need not be connected in time. The TemporalMask D optionally describes a temporal mask giving a set of non-connected sub-intervals in the case that the AV segment is formed from non-connected sub-intervals.

11.6.2.2 AudioVisualSegment DS syntax

```
<!-- ##### -->
<!-- Definition of AudioVisualSegment DS (11.6.2) -->
<!-- ##### -->

<!-- Definition of AudioVisualSegment DS -->
<complexType name="AudioVisualSegmentType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <choice minOccurs="0">
          <element name="MediaTime" type="mpeg7:MediaTimeType"/>
          <element name="TemporalMask" type="mpeg7:TemporalMaskType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SpatialDecomposition"
            type="mpeg7:AudioVisualSegmentSpatialDecompositionType"/>
          <element name="TemporalDecomposition"
            type="mpeg7:AudioVisualSegmentTemporalDecompositionType"/>
          <element name="SpatioTemporalDecomposition"
            type="mpeg7:AudioVisualSegmentSpatioTemporalDecompositionType"/>
          <element name="MediaSourceDecomposition"
            type="mpeg7:AudioVisualSegmentMediaSourceDecompositionType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.6.2.3 AudioVisualSegment DS semantics

Semantics of the AudioVisualSegmentType:

Name	Definition
AudioVisualSegmentType	Describes AV content or a temporal segment of AV content. The temporal localization or composition of the AV segment is optionally described using a choice of MediaTime or TemporalMask. AudioVisualSegmentType extends SegmentType.
MediaTime	Describes the temporal localization of the AV segment using MediaTimeType (optional). The temporal locator describes a temporally connected sub-interval by specifying the start time and the duration of the AV segment. The sub-interval describes a temporal bounding box that may bound a set of individual AV segments. If neither MediaTime nor TemporalMask is described, then the AV segment refers to the entire audio.

Name	Definition
TemporalMask	Describes a temporal mask that defines the temporal composition of the AV segment. (optional). The AV segment is formed from the set of sub-intervals described by the TemporalMask. If absent, the AV segment is composed of the single connected interval defined by MediaTime.
SpatialDecomposition	Describes the spatial decomposition of the AV segment into one or more sub-segments (optional).
TemporalDecomposition	Describes the temporal decomposition of the AV segment into one or more sub-segments (optional).
SpatioTemporalDecomposition	Describes the spatio-temporal decomposition of the AV segment into one or more sub-segments (optional).
MediaSourceDecomposition	Describes the media source decomposition of the AV segment into one or more sub-segments (optional).

11.6.2.4 AudioVisualSegment DS examples (informative)

The following example illustrates the use of the AudioVisualSegment DS for describing an AV segment.

```
<AudioVisualSegment id="DocumentaryAVS">
  <MediaLocator>
    <MediaUri>movie.mpg</MediaUri>
  </MediaLocator>
  <TextAnnotation>
    <FreeTextAnnotation> Documentary about eagles </FreeTextAnnotation>
  </TextAnnotation>
  <MediaTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT1M30S</MediaDuration>
  </MediaTime>
</AudioVisualSegment>
```

11.6.3 Audio-visual segment decomposition tools

11.6.3.1 Introduction

AV segment decomposition tools describe the decomposition of a AV segment into one or more sub-segments in space, time, and media source. The AV segment decomposition tools extend the base segment decomposition tools.

11.6.3.2 Audio-visual segment decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of AudioVisualSegment decomposition DSs (11.6.3) -->
<!-- ##### -->

<!-- Definition of AudioVisualSegmentSpatialDecomposition DS -->
<complexType name="AudioVisualSegmentSpatialDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatialSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="AudioVisualRegion" type="mpeg7:AudioVisualRegionType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

```

        <element name="AudioVisualRegionRef" type="mpeg7:ReferenceType"/>
    </choice>
</extension>
</complexContent>
</complexType>

<!-- Definition of AudioVisualSegmentTemporalDecomposition DS -->
<complexType name="AudioVisualSegmentTemporalDecompositionType">
    <complexContent>
        <extension base="mpeg7:TemporalSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="AudioVisualSegment"
                    type="mpeg7:AudioVisualSegmentType"/>
                <element name="AudioVisualSegmentRef"
                    type="mpeg7:ReferenceType"/>
                <element name="AudioVisualRegion"
                    type="mpeg7:AudioVisualRegionType"/>
                <element name="AudioVisualRegionRef"
                    type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of AudioVisualSegmentSpatioTemporalDecomposition DS -->
<complexType name="AudioVisualSegmentSpatioTemporalDecompositionType">
    <complexContent>
        <extension base="mpeg7:SpatioTemporalSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="AudioVisualRegion"
                    type="mpeg7:AudioVisualRegionType"/>
                <element name="AudioVisualRegionRef"
                    type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of AudioVisualSegmentMediaSourceDecomposition DS -->
<complexType name="AudioVisualSegmentMediaSourceDecompositionType">
    <complexContent>
        <extension base="mpeg7:MediaSourceSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="AudioVisualSegment"
                    type="mpeg7:AudioVisualSegmentType"/>
                <element name="AudioVisualSegmentRef"
                    type="mpeg7:ReferenceType"/>
                <element name="VideoSegment"
                    type="mpeg7:VideoSegmentType"/>
                <element name="VideoSegmentRef"
                    type="mpeg7:ReferenceType"/>
                <element name="StillRegion" type="mpeg7:StillRegionType"/>
                <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
                <element name="AudioSegment"
                    type="mpeg7:AudioSegmentType"/>
                <element name="AudioSegmentRef"
                    type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>

```

</complexType>

11.6.3.3 Audio-visual segment decomposition tools semantics

Semantics of the AudioVisualSegmentSpatialDecompositionType:

<i>Name</i>	<i>Definition</i>
AudioVisualSegmentSpatialDecompositionType	Describes the spatial decomposition of an AV segment into one or more AV regions. AudioVisualSegmentSpatialDecompositionType extends SpatialSegmentDecompositionType.
AudioVisualRegion	Describes an AV region that results from the spatial decomposition (optional).
AudioVisualRegionRef	Describes a reference to the description of an AV region that results from the spatial decomposition (optional).

Semantics of the AudioVisualSegmentTemporalDecompositionType:

<i>Name</i>	<i>Definition</i>
AudioVisualSegmentTemporalDecompositionType	Describes the temporal decomposition of an AV segment into one or more AV segments and AV regions. AudioVisualSegmentTemporalDecompositionType extends TemporalSegmentDecompositionType.
AudioVisualSegment	Describes an AV segment that results from the temporal decomposition (optional).
AudioVisualSegmentRef	Describes a reference to the description of an AV segment that results from the temporal decomposition (optional).
AudioVisualRegion	Describes an AV region that results from the temporal decomposition (optional).
AudioVisualRegionRef	Describes a reference to the description of an AV region that results from the temporal decomposition (optional).

Semantics of the AudioVisualSegmentSpatioTemporalDecompositionType:

<i>Name</i>	<i>Definition</i>
AudioVisualSegmentSpatioTemporalDecompositionType	Describes the spatio-temporal decomposition of an AV segment into one or more AV regions. AudioVisualSegmentSpatioTemporalDecompositionType extends SpatioTemporalSegmentDecompositionType.
AudioVisualRegion	Describes an AV region that results from the spatio-temporal decomposition (optional).
AudioVisualRegionRef	Describes a reference to the description of an AV region that results from the spatio-temporal decomposition (optional).

Semantics of the AudioVisualSegmentMediaSourceDecompositionType:

Name	Definition
AudioVisualSegmentMediaSourceDecompositionType	Describes the media source decomposition of an AV segment into one or more AV segments, video segments, still regions, and audio segments. AudioVisualSegmentMediaSourceDecompositionType extends MediaSourceSegmentDecompositionType.
AudioVisualSegment	Describes an AV segment that results from the media source decomposition (optional).
AudioVisualSegmentRef	Describes a reference to the description of an AV segment that results from the media source decomposition (optional).
VideoSegment	Describes a video segment that results from the media source decomposition (optional).
VideoSegmentRef	Describes a reference to the description of a video segment that results from the media source decomposition (optional).
StillRegion	Describes a still region that results from the media source decomposition (optional).
StillRegionRef	Describes a reference to the description of a still region that results from the media source decomposition (optional).
AudioSegment	Describes an audio segment that results from the media source decomposition (optional).
AudioSegmentRef	Describes a reference to the description of an audio segment that results from the media source decomposition (optional).

11.6.3.4 Audio-visual segment decomposition tools examples (informative)

The following example illustrates the use of the AudioVisualSegmentMediaSourceDecomposition DS in the AudioVisualSegment DS for describing the audio and video constituents of the AV segment "EagleDocumentaryAVS". In this example, the AV segment "EagleDocumentaryAVS" represents the full video sequence that is decomposed into one audio segment, "EagleDocumentaryAS", and one video segment, "EagleDocumentaryVS", corresponding to the audio and video tracks, respectively. Each segment is connected in space and/or time, as applicable.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioVisualType">
      <AudioVisual>
        <MediaLocator>
          <MediaUri>movie.mpg</MediaUri>
        </MediaLocator>
        <TextAnnotation>
          <FreeTextAnnotation> Eagle Documentary </FreeTextAnnotation>
        </TextAnnotation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
        </MediaTime>
      </AudioVisual>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```

        <MediaDuration>PT1M30S</MediaDuration>
    </MediaTime>
    <TemporalDecomposition>
        <AudioVisualSegment id="EagleDocumentaryAVS">
            <TextAnnotation>
                <FreeTextAnnotation> Eagle Documentary </FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
                <MediaTimePoint>T00:00:00</MediaTimePoint>
                <MediaDuration>PT1M30S</MediaDuration>
            </MediaTime>
            <MediaSourceDecomposition gap="false" overlap="false">
                <VideoSegment id="EagleDocumentaryVS">
                    <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>
                </VideoSegment>
                <AudioSegment id="EagleDocumentaryAS">
                    <MediaTime>
                        <MediaTimePoint>T00:00:00</MediaTimePoint>
                        <MediaDuration>PT1M30S</MediaDuration>
                    </MediaTime>
                </AudioSegment>
            </MediaSourceDecomposition>
        </AudioVisualSegment>
    </TemporalDecomposition>
</AudioVisual>
</MultimediaContent>
</Description>
</Mpeg7>

```

11.6.4 AudioVisualRegion DS

11.6.4.1 Introduction

The AudioVisualRegion DS describes AV content or an arbitrary spatio-temporal region of AV content. The audio from the region belongs to a specified temporal interval, and the video belongs to a specified spatio-temporal region. The video part of the AV region need not be connected in space or time. The SpatioTemporalMask D optionally describes a spatio-temporal mask giving a set of non-connected sub-regions in the case that the video part of the AV region is formed from non-connected sub-regions. The audio part of the AV region also need not be connected in time. The TemporalMask D optionally describes a temporal mask giving a set of non-connected sub-intervals in the case that the audio part is formed from non-connected sub-intervals.

11.6.4.2 AudioVisualRegion DS syntax

```

<!-- ##### -->
<!-- Definition of AudioVisualRegion DS (11.6.4) -->
<!-- ##### -->

<!-- Definition of AudioVisualRegion DS -->

```

```

<complexType name="AudioVisualRegionType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <choice minOccurs="0">
          <element name="VisualSpatioTemporalLocator"
            type="mpeg7:SpatioTemporalLocatorType"/>
          <element name="VisualSpatioTemporalMask"
            type="mpeg7:SpatioTemporalMaskType"/>
        </choice>
        <choice minOccurs="0">
          <element name="AudioMediaTime" type="mpeg7:MediaTimeType"/>
          <element name="AudioTemporalMask" type="mpeg7:TemporalMaskType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SpatialDecomposition"
            type="mpeg7:AudioVisualRegionSpatialDecompositionType"/>
          <element name="TemporalDecomposition"
            type="mpeg7:AudioVisualRegionTemporalDecompositionType"/>
          <element name="SpatioTemporalDecomposition"
            type="mpeg7:AudioVisualRegionSpatioTemporalDecompositionType"/>
          <element name="MediaSourceDecomposition"
            type="mpeg7:AudioVisualRegionMediaSourceDecompositionType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

11.6.4.3 AudioVisualRegion DS semantics

Semantics of the AudioVisualRegionType:

Name	Definition
AudioVisualRegionType	Describes AV content or an arbitrary region of AV content. The audio from the region belongs to a specified temporal interval, and the video belongs to a specified spatio-temporal region. The audio part of the AV region need not be connected in time, and the video part of the AV region need not be connected in space or time. The spatio-temporal localization or composition of the video part of the AV region is optionally described using a choice of SpatioTemporalLocator or SpatioTemporalMask. The temporal localization or composition of the audio part of the AV region is optionally described using a choice of MediaTime or TemporalMask. AudioVisualRegionType extends SegmentType.
VisualSpatioTemporalLocator	Describes the spatio-temporal localization of the video part of the AV region using SpatioTemporalLocatorType (optional). The spatio-temporal locator describes a spatio-temporally connected sub-region that includes the spatio-temporal components of the video part of the AV region. The SpatioTemporalLocatorType is defined in ISO/IEC 15938-3.
VisualSpatioTemporalMask	Describes a spatio-temporal mask that defines the spatio-temporal composition of the video part of the AV region. (optional). The region is formed from the set of sub-regions described by the VisualSpatioTemporalMask. If absent, the region is composed of the single connected region defined by the VisualSpatioTemporalLocator.

Name	Definition
AudioMediaTime	Describes the temporal localization of the audio part of the AV region using MediaTimeType (optional). The temporal locator describes a temporally connected sub-interval by specifying the start time and the duration of the audio segment. The sub-interval describes a temporal bounding box that may bound a set of individual segments. If neither AudioMediaTime nor AudioTemporalMask is described, then the segment refers to the entire audio.
AudioTemporalMask	Describes a temporal mask that defines the temporal composition of the audio part of the AV region (optional). The audio segment is formed from the set of sub-intervals described by the AudioTemporalMask. If absent, the segment is composed of the single connected interval defined by AudioMediaTime.
SpatialDecomposition	Describes the spatial decomposition of the AV region into one or more sub-segments (optional).
TemporalDecomposition	Describes the temporal decomposition of the AV region into one or more sub-segments (optional).
SpatioTemporalDecomposition	Describes the spatio-temporal decomposition of the AV region into one or more sub-segments (optional).
MediaSourceDecomposition	Describes the media source decomposition of the AV region into one or more sub-segments (optional).

11.6.4.4 AudioVisualRegion DS examples (informative)

The following example illustrates the use of the AudioVisualRegion DS for describing an AV Region. In this example, the AV region id = "NarratorAVR" represents the spatio-temporal region of the video and the temporal segment of the audio corresponding to the object "Narrator". The VisualSpatioTemporalLocator and the AudioMediaTime locate the AV region in the video and the audio, respectively.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioVisualType">
      <AudioVisual>
        <MediaLocator>
          <MediaUri>movie.mpg</MediaUri>
        </MediaLocator>
        <TextAnnotation>
          <FreeTextAnnotation> Eagle Documentary </FreeTextAnnotation>
        </TextAnnotation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT1M30S</MediaDuration>
        </MediaTime>
        <SpatioTemporalDecomposition>
          <AudioVisualRegion id="NarratorAVR">
            <TextAnnotation>
              <FreeTextAnnotation> Video and audio of narrator
            </FreeTextAnnotation>
            </TextAnnotation>
            <VisualSpatioTemporalLocator>
              <FigureTrajectory type="rectangle">

```

```

    <MediaTime>
      <MediaTimePoint>T00:00:15</MediaTimePoint>
      <MediaDuration>PT1M15S</MediaDuration>
    </MediaTime>
    <Vertex>
      <WholeInterval>
        <MediaDuration>PT0S</MediaDuration>
      </WholeInterval>
      <InterpolationFunctions>
        <KeyValue type="startPoint">0.1</KeyValue>
        <KeyValue type="startPoint">0.2</KeyValue>
      </InterpolationFunctions>
    </Vertex>
    <Vertex>
      <WholeInterval>
        <MediaDuration>PT1S</MediaDuration>
      </WholeInterval>
      <InterpolationFunctions>
        <KeyValue type="startPoint">0.3</KeyValue>
        <KeyValue type="startPoint">0.4</KeyValue>
      </InterpolationFunctions>
    </Vertex>
    <Vertex>
      <WholeInterval>
        <MediaDuration>PT2S</MediaDuration>
      </WholeInterval>
      <InterpolationFunctions>
        <KeyValue type="startPoint">0.5</KeyValue>
        <KeyValue type="startPoint">0.6</KeyValue>
      </InterpolationFunctions>
    </Vertex>
    <!-- more elements here -->
  </FigureTrajectory>
</VisualSpatioTemporalLocator>
<AudioMediaTime>
  <MediaTimePoint>T00:00:00</MediaTimePoint>
  <MediaDuration>PT1M30S</MediaDuration>
</AudioMediaTime>
</AudioVisualRegion>
</SpatioTemporalDecomposition>
</AudioVisual>
</MultimediaContent>
</Description>
</Mpeg7>

```

11.6.5 Audio-visual region decomposition tools

11.6.5.1 Introduction

The AV region segment decomposition tools describe the decomposition of an AV region into one or more sub-segments in space, time, and media source. The AV region decomposition tools extend the base segment decomposition tools.

11.6.5.2 Audio-visual region decomposition tools syntax

```

<!-- ##### -->
<!-- Definition of AudioVisualRegion decomposition DSs (11.6.5) -->
<!-- ##### -->

```



```

<!-- Definition of AudioVisualRegionSpatialDecomposition DS -->
<complexType name="AudioVisualRegionSpatialDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatialSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="AudioVisualRegion"
          type="mpeg7:AudioVisualRegionType"/>
        <element name="AudioVisualRegionRef"
          type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of AudioVisualSegmentTemporalDecomposition DS -->
<complexType name="AudioVisualRegionTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:TemporalSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="AudioVisualRegion"
          type="mpeg7:AudioVisualRegionType"/>
        <element name="AudioVisualRegionRef"
          type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of AudioVisualSegmentSpatioTemporalDecomposition DS -->
<complexType name="AudioVisualRegionSpatioTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatioTemporalSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="AudioVisualRegion"
          type="mpeg7:AudioVisualRegionType"/>
        <element name="AudioVisualRegionRef"
          type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of AudioVisualSegmentMediaSourceDecomposition DS -->
<complexType name="AudioVisualRegionMediaSourceDecompositionType">
  <complexContent>
    <extension base="mpeg7:MediaSourceSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="AudioVisualRegion"
          type="mpeg7:AudioVisualRegionType"/>
        <element name="AudioVisualRegionRef"
          type="mpeg7:ReferenceType"/>
        <element name="VideoSegment"
          type="mpeg7:VideoSegmentType"/>
        <element name="VideoSegmentRef"
          type="mpeg7:ReferenceType"/>
        <element name="MovingRegion"
          type="mpeg7:MovingRegionType"/>
        <element name="MovingRegionRef"
          type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

```

        <element name="StillRegion" type="mpeg7:StillRegionType"/>
        <element name="StillRegionRef" type="mpeg7:ReferenceType"/>
        <element name="AudioSegment"
            type="mpeg7:AudioSegmentType"/>
        <element name="AudioSegmentRef"
            type="mpeg7:ReferenceType"/>
    </choice>
</extension>
</complexContent>
</complexType>

```

11.6.5.3 Audio-visual region decomposition tools semantics

Semantics of the AudioVisualRegionSpatialDecompositionType:

Name	Definition
AudioVisualRegionSpatialDecompositionType	Describes the spatial decomposition of an AV region into one or more AV regions. AudioVisualRegionSpatialDecompositionType extends SpatialSegmentDecompositionType.
AudioVisualRegion	Describes an AV region that results from the spatial decomposition (optional).
AudioVisualRegionRef	Describes a reference to the description of an AV region that results from the spatial decomposition (optional).

Semantics of the AudioVisualRegionTemporalDecompositionType:

Name	Definition
AudioVisualRegionTemporalDecompositionType	Describes the temporal decomposition of an AV region into one or more AV regions. AudioVisualRegionTemporalDecompositionType extends TemporalSegmentDecompositionType.
AudioVisualRegion	Describes an AV region that results from the temporal decomposition (optional).
AudioVisualRegionRef	Describes a reference to the description of an AV region that results from the temporal decomposition (optional).

Semantics of the AudioVisualRegionSpatioTemporalDecompositionType:

Name	Definition
AudioVisualRegionSpatioTemporalDecompositionType	Describes the spatio-temporal decomposition of an AV region into one or more AV regions. AudioVisualRegionTemporalDecompositionType extends

Name	Definition
	SpatioTemporalSegmentDecompositionType.
AudioVisualRegion	Describes an AV region that results from the spatio-temporal decomposition (optional).
AudioVisualRegionRef	Describes a reference to the description of an AV region that results from the spatio-temporal decomposition (optional).

Semantics of the AudioVisualRegionMediaSourceDecompositionType:

Name	Definition
AudioVisualRegionMediaSourceDecompositionType	Describes the media source decomposition of an AV region into one or more AV regions, video segments, moving regions, still regions, and audio segments. AudioVisualRegionMediaSourceDecompositionType extends MediaSourceSegmentDecompositionType.
AudioVisualRegion	Describes an AV region that results from the media source decomposition (optional).
AudioVisualRegionRef	Describes a reference to the description of an AV region that results from the media source decomposition (optional).
VideoSegment	Describes a video segment that results from the media source decomposition (optional).
VideoSegmentRef	Describes a reference to the description of a video segment that results from the media source decomposition (optional).
MovingRegion	Describes a moving region that results from the media source decomposition (optional).
MovingRegionRef	Describes a reference to the description of a moving region that results from the media source decomposition (optional).
StillRegion	Describes a still region that results from the media source decomposition (optional).
StillRegionRef	Describes a reference to the description of a still region that results from the media source decomposition (optional).
AudioSegment	Describes an audio segment that results from the media source decomposition (optional).
AudioSegmentRef	Describes a reference to the description of an audio segment that results from the media source decomposition (optional).

11.6.5.4 Audio-visual region decomposition tools examples (informative)

The following example illustrates the use of the AudioVisualRegionMediaSourceDecomposition DS in the AudioVisualRegion DS for describing the video and audio constituents of the AV region "NarratorAVR" in 11.6.4.4. In this example, the AV region "NarratorAVR" is decomposed into the moving region "NarratorMR" and the audio segment "NarratorAS" corresponding to the video and audio data of the object "Narrator", respectively. Each segment is connected in space and/or time, as applicable.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioVisualType">
      <AudioVisual>
        <MediaLocator>
          <MediaUri>movie.mpg</MediaUri>
        </MediaLocator>
        <TextAnnotation>
          <FreeTextAnnotation> Eagle Documentary </FreeTextAnnotation>
        </TextAnnotation>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT1M30S</MediaDuration>
        </MediaTime>
        <SpatioTemporalDecomposition>
          <AudioVisualRegion id="NarratorAVR">
            <TextAnnotation>
              <FreeTextAnnotation> Video and audio of narrator
            </FreeTextAnnotation>
            </TextAnnotation>
            <MediaSourceDecomposition gap="false" overlap="false">
              <MovingRegion id="NarratorMR">
                <TextAnnotation>
                  <FreeTextAnnotation> Moving region of narrator
                </FreeTextAnnotation>
                </TextAnnotation>
              </MovingRegion>
              <AudioSegment id="NarratorAS">
                <MediaTime>
                  <MediaTimePoint>T00:00:00</MediaTimePoint>
                  <MediaDuration>PT01M30S</MediaDuration>
                </MediaTime>
              </AudioSegment>
            </MediaSourceDecomposition>
          </AudioVisualRegion>
        </SpatioTemporalDecomposition>
      </AudioVisual>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

11.7 Multimedia segment description tools

11.7.1 Introduction

This subclause specifies tools for describing segments of multimedia content and their structural decompositions. The following table summarizes the tools specified in this subclause. Figure 43 shows the organization of the tools for describing segments of multimedia content.

Table 22 — Overview of the multimedia segment description tools

Tool	Functionality
MultimediaSegment DS	Describes multimedia content and segments of multimedia content. The MultimediaSegment DS extends the Segment DS.
Multimedia decomposition tools	These tools describe decompositions of multimedia segments in media source. These tools extend the base segment decomposition tools. The following multimedia segment decomposition tools are defined: MultimediaSegmentMediaSourceDecomposition DS: describes a media source decomposition of a multimedia segment.

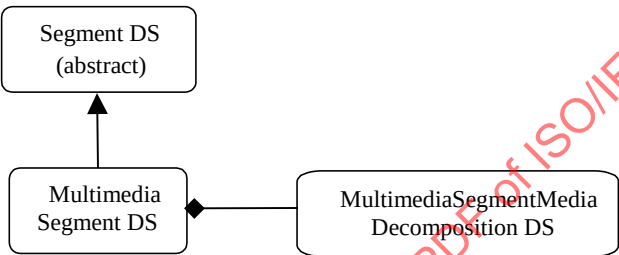


Figure 43 — Overview of the tools for describing segments of multimedia content

11.7.2 MultimediaSegment DS

11.7.2.1 Introduction

The MultimediaSegment DS describes multimedia content or a segment of multimedia content, which contain audio, video, and possibly other multimedia content. The MultimediaSegment DS extends the Segment DS.

11.7.2.2 MultimediaSegment DS syntax

```
<!-- ##### -->
<!-- Definition of MultimediaSegment DS (11.7.2) -->
<!-- ##### -->

<!-- Definition of MultimediaSegment DS -->
<complexType name="MultimediaSegmentType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <element name="MediaSourceDecomposition"
          type="mpeg7:MultimediaSegmentMediaSourceDecompositionType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.7.2.3 MultimediaSegment DS semantics

Semantics of the MultimediaSegmentType:

<i>Name</i>	<i>Definition</i>
MultimediaSegmentType	Describes multimedia content or a segment of multimedia content. MultimediaSegmentType extends SegmentType.
MediaSourceDecomposition	Describes the media source decomposition of the multimedia segment into one or more sub-segments (optional).

11.7.2.4 MultimediaSegment DS examples (informative)

The following examples illustrate the use of the MultimediaSegment DS for describing multimedia content.

```
<MultimediaSegment>
  <TextAnnotation>
    <FreeTextAnnotation> Multimedia content </FreeTextAnnotation>
  </TextAnnotation>
</MultimediaSegment>
```

11.7.3 Multimedia segment decomposition tools**11.7.3.1 Introduction**

The multimedia segment decomposition tools extend the base segment decomposition tools. The multimedia segment decomposition tools describe the decomposition of a multimedia segment into one or more sub-segments in media source.

11.7.3.2 Multimedia segment decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of MultimediaSegmentMediaSourceDecomposition DS (11.7.3) -->
<!-- ##### -->

<!-- Definition of MultimediaSegmentMediaSourceDecomposition DS -->
<complexType name="MultimediaSegmentMediaSourceDecompositionType">
  <complexContent>
    <extension base="mpeg7:MediaSourceSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="Segment" type="mpeg7:SegmentType"/>
        <element name="SegmentRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

11.7.3.3 Multimedia segment decomposition tools semantics

Semantics of the MultimediaSegmentMediaSourceDecompositionType:

Name	Definition
MultimediaSegmentMediaSourceDecompositionType	Describes the media source decomposition of a multimedia segment into one or more segments. MultimediaSegmentMediaSourceDecompositionType extends MediaSourceSegmentDecompositionType.
Segment	Describes a segment that results from the media source decomposition (optional).
SegmentRef	Describes a reference to the description of a segment that results from the media source decomposition (optional).

11.7.3.4 Multimedia segment decomposition tools examples (informative)

The following examples illustrate the use of the MultimediaSegmentMediaSourceDecomposition DS for describing the decomposition of multimedia content into its video, audio, and image components.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaType">
      <Multimedia>
        <MediaSourceDecomposition gap="false" overlap="false">
          <Segment xsi:type="VideoSegmentType">
            <TextAnnotation>
              <FreeTextAnnotation> Chase video </FreeTextAnnotation>
            </TextAnnotation>
            <MediaTime>
              <MediaTimePoint>T00:00:00</MediaTimePoint>
              <MediaDuration>PT0M15S</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>
          </Segment>
          <Segment xsi:type="AudioSegmentType">
            <TextAnnotation>
              <FreeTextAnnotation> Chase audio </FreeTextAnnotation>
            </TextAnnotation>
          </Segment>
          <Segment xsi:type="StillRegionType">
            <TextAnnotation>
              <FreeTextAnnotation> Chase image </FreeTextAnnotation>
            </TextAnnotation>
          </Segment>
        </MediaSourceDecomposition>
      </Multimedia>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

11.8 Ink segment description tools

11.8.1 Introduction

This subclause specifies tools for describing segments of ink content and their attributes and structural decompositions. The following table summarizes the tools specified in this subclause. Figure 44 shows the organization of the tools for describing segments of ink content.

Table 23 — Overview of the ink segment description tools

<i>Tool</i>	<i>Functionality</i>
InkSegment DS	This tool describes a spatio-temporal segment of ink data. The InkSegment DS extends the Segment DS.
InkMediaInformation DS	This tool describes specific media and creation information related to an ink segment. This tool applies to the InkSegment DS.
HandWritingRecogInformation DS	This tool describes information about a hand-writing recognizer. This tool applies to the InkSegment DS.
HandWritingRecogResult DS	This tool describes hand-writing recognition results. This tool applies to the InkSegment DS.
Ink segment decomposition tools	These tools describe decompositions of ink segments in space and time. These tools extend the base segment decomposition tools. The following ink segment decomposition tools are defined: • InkSegmentSpatialDecomposition DS: describes a spatial decomposition of an ink segment. • InkSegmentTemporalDecomposition DS: describes a temporal decomposition of an ink segment.

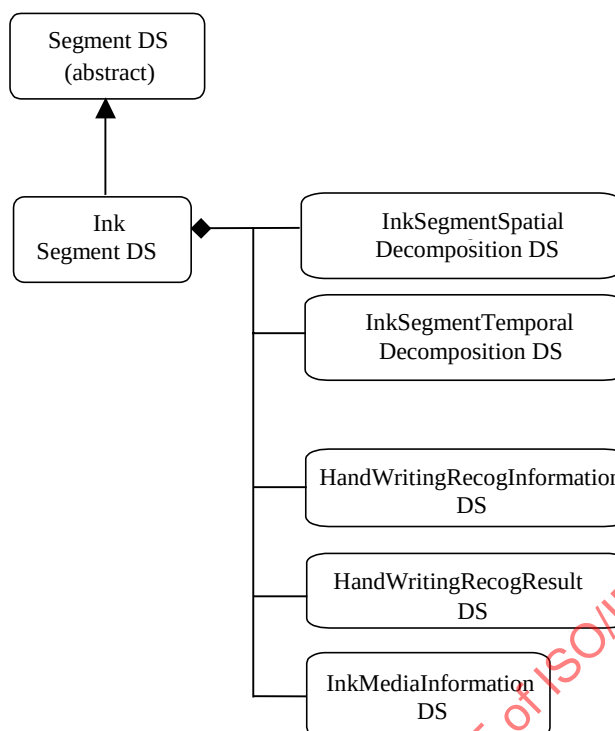


Figure 44 — Overview of the tools for describing segments of ink content

Ink content is created from various pen devices and other ink applications (e.g., electronic whiteboard applications, form filling and ink messaging). An ink document may or may not be in electronic form. For example, paper work such as a book or a painting may contain afterward footnotes or added ink strokes. An ink document consists of ink content and metastrokes. Ink content and metastrokes consists of sets of ink strokes. Ink strokes are ink data points between the pen down and the pen up states or equivalent. The content ink strokes are intended to be recorded in the ink content. The meta ink strokes (or metastrokes) are intended to express information about content ink strokes or other visual content (e.g., images and video sequences) such as personal marks or keyword gesture commands provided by the content creator or content reviewers. Examples of content ink strokes and metastrokes are shown in Figure 45.

Figure 45 illustrates the relationships of the tools for describing segments of ink. The ink data files usually record only the content ink strokes (label 1). The meta ink strokes are usually stored independently of content ink strokes. Furthermore, the meta ink strokes need not be displayed during the ink content playback. The InkSegment DS describes both types of ink segments, content ink strokes and metastrokes (label 2) using the InkMediaInformation DS, among others (label 3). A handwriting engine (HWR Engine) could process the ink data file to recognize of hand-written characters and symbols (label 5). The HandWritingRecogInformation DS and the HandWritingRecogResult DS describe the hand-writing recognizer and results of the recognition process (label 5). Finally, the SceneGraphMask, MediaSpaceMask, OrderedGroupDataSetMask and the TemporalMask D localize segments within ink data files.

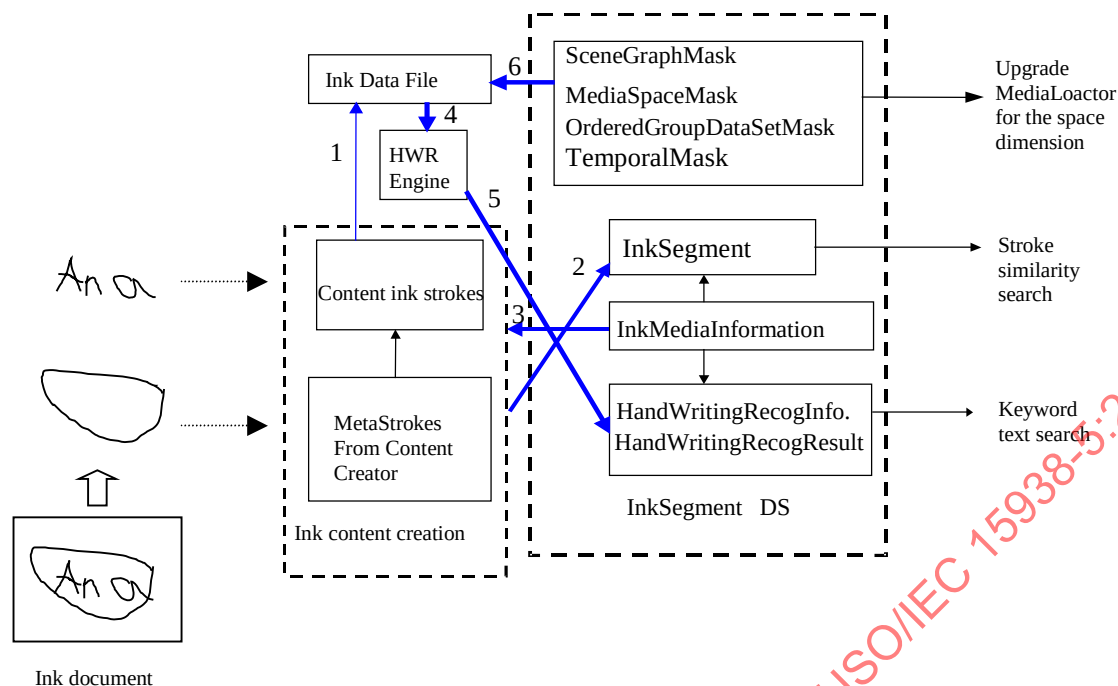


Figure 45 — Illustration of the relationships of the tools for describing segments of ink data

11.8.2 InkSegment DS

11.8.2.1 Introduction

The InkSegment DS describes a spatial and/or temporal interval or segment of ink data, which corresponds to a set of content ink data points and meta ink data points. The InkSegment DS extends the Segment DS. The ink segment need not be connected in space and/or time. The TemporalMask D, SceneGraphMask D, MediaSpaceMask D, and OrderedGroupDataSetMask D describe the composition of individual spatial and/or temporal components that forms a non-connected ink segment. The InkSegment DS describes the visual feature, localization, media, creation, and hand-writing recognition information, in addition to the other properties by the Segment DS.

11.8.2.2 InkSegment DS syntax

```
<!-- ##### -->
<!-- Definition of InkSegment DS (11.8.2) -->
<!-- ##### -->

<!-- Definition of InkSegment DS -->
<complexType name="InkSegmentType">
  <complexContent>
    <extension base="mpeg7:SegmentType">
      <sequence>
        <element name="InkMediaInformation"
          type="mpeg7:InkMediaInformationType" minOccurs="0"/>
        <element name="MediaTime" type="mpeg7:MediaTimeType" minOccurs="0"/>
        <element name="TemporalMask" type="mpeg7:TemporalMaskType"
          minOccurs="0"/>
        <element name="SceneGraphMask" type="mpeg7:SceneGraphMaskType"
          minOccurs="0"/>
        <element name="MediaSpaceMask" type="mpeg7:MediaSpaceMaskType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

        minOccurs="0"/>
<element name="OrderedGroupDataSetMask"
        type="mpeg7:OrderedGroupDataSetMaskType"
        minOccurs="0"/>
<choice minOccurs="0" maxOccurs="unbounded">
    <element name="HandWritingRecogInformation"
        type="mpeg7:HandWritingRecogInformationType"/>
    <element name="HandWritingRecogResult"
        type="mpeg7:HandWritingRecogResultType"/>
</choice>
<choice minOccurs="0" maxOccurs="unbounded">
    <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
    <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
    <element name="VisualTimeSeriesDescriptor"
        type="mpeg7:VisualTimeSeriesType"/>
</choice>
<choice minOccurs="0" maxOccurs="unbounded">
    <element name="SpatialDecomposition"
        type="mpeg7:InkSegmentSpatialDecompositionType"/>
    <element name="TemporalDecomposition"
        type="mpeg7:InkSegmentTemporalDecompositionType"/>
</choice>
</sequence>
<attribute name="type" use="optional" default="content">
    <simpleType>
        <restriction base="NMTOKEN">
            <enumeration value="content"/>
            <enumeration value="meta"/>
            <enumeration value="mixed"/>
        </restriction>
    </simpleType>
</attribute>
</extension>
</complexContent>
</complexType>

```

11.8.2.3 InkSegment DS semantics

Semantics of the InkSegmentType:

Name	Definition
InkSegmentType	Describes a spatio-temporal segment of ink data. The ink segment need not be connected in space or time. InkSegmentType extends SegmentType. Ink data is composed of ink strokes, which are ink data points between the pen down and the pen up states or the equivalent. The content ink strokes are intended to be recorded in the ink document. The meta ink strokes (or metastrokes) are intended to express information about content ink strokes or other visual content (e.g., images or video sequences) such as personal marks or gesture commands. The meta ink strokes may be stored independently of content ink strokes. Furthermore, the meta ink strokes need not be displayed during the ink content playback.

<i>Name</i>	<i>Definition</i>
InkMediaInformation	Describes specific media and creation information related to the ink segment such as the input device, the layout of the writing regions or fields in a form, the brush, the overlay of ink segment over other visual content, and the writing style (optional).
MediaTime	Describes the temporal localization of the ink segment by specifying the start time and the duration of the ink segment (optional). If the ink segment is non-connected, the duration should be equal to the duration of the smallest connected temporal interval that includes the ink segment.
TemporalMask	Describes the composition of temporally non-overlapping, connected components that form a non-connected ink segment (optional). If absent, the segment is composed of the single connected interval in time defined by the MediaTime element. If present, the segment refers to the set of non-overlapping sub-intervals in time defined by the TemporalMask element.
SceneGraphMask	Describes the composition of non-overlapping sub-graphs in a scene graph that form an ink segment (optional). If the scene, media space, and ordered group mask elements are absent, the segment is composed of the interval(s) in time defined by the MediaTime and the TemporalMask elements. If more than one mask is present, the ink segment is the intersection of the ink data located by each mask (the TemporalMask element, the SceneGraphMask element, the MediaSpaceMask element, and the OrderedGroupDataSetMask element).
MediaSpaceMask	Describes the composition of non-overlapping binary and text data intervals in a file that form an ink segment (optional). If the scene, media space, and ordered group mask elements are absent, the segment is composed of the interval(s) in time defined by the MediaTime and the TemporalMask elements. If more than one mask is present, the ink segment is the intersection of the ink data located by each mask (the TemporalMask element, the SceneGraphMask element, the MediaSpaceMask element, and the OrderedGroupDataSetMask element).
OrderedGroupDataSetMask	Describes the composition of non-overlapping subsets in an ordered group dataset that form an ink segment (optional). If the scene, media space, and ordered group mask elements are absent, the segment is composed of the interval(s) in time defined by the MediaTime and the TemporalMask elements. If more than one mask is present, the ink segment is the intersection of the ink data located by each mask (the TemporalMask element, the SceneGraphMask element, the MediaSpaceMask element, and the OrderedGroupDataSetMask element).
HandWritingRecogInformation	Describes information about a hand-writing recognizer (optional).
HandWritingRecogResult	Describes hand-writing recognition results of the ink segment (optional).
VisualDescriptor	Describes a visual feature of the ink segment using a visual descriptor (optional). The specific VisualDTypes are defined in ISO/IEC 15938-3.
VisualDescriptionScheme	Describes visual features of the ink segment using a visual description scheme (optional). The specific VisualDSTypes are defined in ISO/IEC 15938-3.

Name	Definition
VisualTimeSeriesDescriptor	Describes a temporal sequence of visual features in the ink segment (optional). The VisualTimeSeriesDescriptor applies only in the case of a connected ink segment. The VisualTimeSeriesDescriptorType is defined in ISO/IEC 15938-3.
SpatialDecomposition	Describes a spatial decomposition of the ink segment into one or more sub-segments (optional).
TemporalDecomposition	Describes a temporal decomposition of the ink segment into one or more sub-segments (optional).
type	Indicates the type of the ink segment. The types of ink segments are defined as follows: • <i>content</i> – The ink segment is a content ink segment, which is a set of content ink strokes. For example, the word "Ana" in the ink document shown in Figure 45 is a content ink segment. • <i>meta</i> – The ink segment is a meta ink segment, which is a set of meta ink strokes. For example, the keyword gesture (i.e., circle) in the ink document shown in Figure 45 is a meta ink segment. Other examples of meta ink segments are check and star marks. • <i>mixed</i> – The ink segment is a mixed ink segment, which is a set of meta and content ink strokes. For example, the keyword gesture (i.e., circle) together with the word "Ana" in the ink document shown in Figure 45 is a mixed ink segment. This attribute value is "content" by default.

11.8.2.4 Ink segment constraints

11.8.2.4.1 Types of ink segments

There are three different types of ink segments based on the type of strokes of data points they contain: content, meta, and mixed ink segments (see type attribute). Content ink segments are sets of points of content strokes; meta ink segments are sets of points of meta strokes; finally, mixed ink segments are sets of points of both content and meta ink strokes. Meta ink segments express semantic gesture or command information about ink content or other visual content.

Ink content can overlay (e.g., form filling, highlight, command, and annotation) over any type of visual content. In these cases, the OverlaidMedia element in the InkMediaInformation DS describes the association and mapping between the ink data and the overlaid media. The relation *annotates* may describe that an ink segment annotates or describes the content of another segments including other ink segments, still regions, and video segments, among others.

During the creating process, the ink content creator may naturally divide the ink strokes into groups of strokes (e.g., pages, paragraphs, sentences, and commands). The ink segment decomposition tools may describe this creator-defined stroke grouping structure; the StructuralUnit element (ControlledTermUse datatype) inherited from the Segment DS can describe the structural unit of an ink segment including its definition (e.g., "page", "paragraph", "sentence", and "command name"). Param elements in the InkMediaInformation DS describe the parameters associated with a command.

11.8.2.4.2 Space and time coordinates

Each data point in ink content has associated discrete or continuous spatial and/or temporal information in a spatial and/or temporal coordinate system. When both spatial and temporal information is available for the points in a ink content, there are no restrictions between the spatial and temporal dimensions of the data points. For example, the spatial and temporal coordinate system of an ink document may correspond to an arbitrary sampling or quantization of the original spatial and temporal space in which the ink document was created. As a result, two data points can share the same temporal and/or spatial information. The spatial and/or temporal coordinates of an ink content can be defined based on the creation or display processes. Examples are provided in the following subclause.

11.8.2.5 InkSegment DS examples (informative)

The following example illustrates the use of the InkSegment DS for describing the ink document shown in Figure 45. The ink document in Figure 45 consists of the word "Ana" surrounded by a circle keyword gesture. The strokes corresponding to the word "Ana" are a content ink segment; the strokes corresponding to the keyword gesture are a meta ink segment; finally, the full ink document is a mixed ink segment. In this example, the ink segments "ElecDoc1", "WordAna", and "KeywordGesture" represent the mixed ink segment, the content ink segment, and the meta ink segment, respectively. The ink segments "WordAna" and "KeywordGesture" are the result of the temporal decomposition of ink segment "ElecDoc1" because they have different temporal information.

In this example, the ink capture device records the temporal information of the ink data points with very low frequency so all the ink data points of ink segment "WordAna" carry the same temporal information and the ink segment "WordAna" can be spatially decomposed into ink segments that represent the different letters of the word, e.g., ink segment "LetterA".

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="InkContentType">
      <InkContent id="ElecDoc1" type="mixed">
        <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>
        <TemporalDecomposition>
          <InkSegment id="WordAna" type="content">
            <MediaTime>
              <MediaRelTimePoint
mediaTimeBase="../../../../MediaLocator[1]">PT0S</MediaRelTimePoint>
              <MediaDuration>PT3S</MediaDuration>
            </MediaTime>
            <HandWritingRecogResult>
              <Quality>0.6</Quality>
              <Result score="0.7">
                <Text> Ana </Text>
              </Result>
              <Result score="0.54">
                <Text> Ann </Text>
              </Result>
            </HandWritingRecogResult>
            <SpatialDecomposition>
              <InkSegment id="LetterA" type="content">
                <OrderedGroupDataSetMask>
```

```

        <SubInterval setRef="#WordAna" startComponent="1731"
            startUnit="36" endComponent="1742" endUnit="47"/>
    </OrderedGroupDataSetMask>
    <HandWritingRecogResult>
        <Quality>0.4</Quality>
        <Result score="0.54">
            <Text> A </Text>
        </Result>
    </HandWritingRecogResult>
</InkSegment>
</SpatialDecomposition>
</InkSegment>
<InkSegment id="KeywordGesture" type="meta">
    <MediaTime>
        <MediaRelTimePoint
            mediaTimeBase="../../../../MediaLocator[1]">PT3S</MediaRelTimePoint>
        <MediaDuration>PT2S</MediaDuration>
    </MediaTime>
</InkSegment>
</TemporalDecomposition>
</InkContent>
</MultimediaContent>
</Description>
</Mpeg7>

```

It is believed that BSE is caused by a natural protein - called a prion - which then folded up in the wrong way causes other similar proteins to change into the same shape. When BSE was first identified scientists examined a variety of possible causes. In 1988 it was determined that the disease was caused by an agent similar to the one which causes the sheep disease scrapie.

```

.HIERARCHY PARAGRAPH LINE WORD CHARACTER STROKE
.SEGMENT LINE 1699:43-1748:90 OK "causes. In 1988 it was..."
.SEGMENT WORD 1731:36-1742:47 OK "determined"
.SEGMENT CHAR 1732:3-1732:42 OK "d"

```

pen-stream number sample number

Figure 46 — Example of an ink document

The following example illustrates the use of the InkSegment DS to describe the ink document shown in Figure 46. The last four lines in Figure 46 are part of the ink data file and describe the structure of the ink data points in segments. For example, the third to last line describes a line in the ink data file that consists of the ink data points from stroke number 1699 point 43 to stroke number 1748 point 90 with the following recognized text "cause. In 1988 it was ...".


```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="InkContentType">
      <InkContent id="Paragraph-1">
        <MediaLocator>
          <MediaUri>http://www.mpeg7.org/demo.stk</MediaUri>
        </MediaLocator>
        <SpatialDecomposition gap="true" overlap="false">
          <InkSegment id="Line-7">
            <InkMediaInformation>
              <InputDevice resolutionS="120">
                <Device>
                  <Name xml:lang="en">PenDevice</Name>
                </Device>
              </InputDevice>
            </InkMediaInformation>
            <OrderedGroupDataSetMask>
              <SubInterval setRef="#Paragraph-1" startComponent="1699"
                startUnit="43" endComponent="1748" endUnit="90"/>
            </OrderedGroupDataSetMask>
            <HandWritingRecogResult>
              <Quality>0.9</Quality>
              <Result score="1.0">
                <Text> causes. In 1988 it was determined that the </Text>
              </Result>
            </HandWritingRecogResult>
          </SpatialDecomposition gap="true" overlap="false">
            <InkSegment id="WORD-6">
              <OrderedGroupDataSetMask>
                <SubInterval setRef="#Paragraph-1"
                  startComponent="1731" startUnit="36"
                  endComponent="1742" endUnit="47"/>
              </OrderedGroupDataSetMask>
              <HandWritingRecogResult>
                <Quality>0.9</Quality>
                <Result score="1.0">
                  <Text> determined </Text>
                </Result>
              </HandWritingRecogResult>
            </InkSegment>
          </SpatialDecomposition>
        </InkSegment>
      </SpatialDecomposition>
    </InkContent>
  </MultimediaContent>
</Description>
</Mpeg7>

```

The following example illustrates the use of the StructuralUnit element in the InkSegment DS for describing the grouping of ink data points into pages, paragraph and commands, as defined by the creator. The root ink segment contains one content ink segment and one meta ink segment. The content ink segment is a page and is composed of two other content ink segments that are paragraphs. This example also illustrate the use of the Param element and the OverlaidMedia element in InkMediaInformation DS for describing the two parameters and the operand of the command, respectively. The command mails the ink segment "data" from foo@columbia.edu to foo@mpeg.nist.gov.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="InkContentType">

```



```

<InkContent type="mixed">
  <TemporalDecomposition>
    <InkSegment id="data" type="content">
      <StructuralUnit href="urn:example:acs">
        <Name>Page</Name>
      </StructuralUnit>
      <TemporalDecomposition>
        <InkSegment>
          <StructuralUnit href="urn:example:acs">
            <Name>Paragraph</Name>
          </StructuralUnit>
          <MediaTime>
            <MediaTimePoint>T00:00:00</MediaTimePoint>
            <MediaDuration>PT1M30S</MediaDuration>
          </MediaTime>
        </InkSegment>
        <InkSegment>
          <StructuralUnit href="urn:example:acs">
            <Name>Paragraph</Name>
          </StructuralUnit>
          <MediaTime>
            <MediaTimePoint>T00:01:30</MediaTimePoint>
            <MediaDuration>PT1M04S</MediaDuration>
          </MediaTime>
        </InkSegment>
      </TemporalDecomposition>
    </InkSegment>
    <InkSegment type="meta">
      <StructuralUnit href="urn:example:acs">
        <Name>MAIL command</Name>
        <Definition>
MAIL command is used to mail the content specified by the
OverlaidMedia element in the InkMediaInformation DS. There are two
Associated parameterName "From" and "To".
The associated parameterValues are to specify the sender and
receiver email address respectively.
        </Definition>
      </StructuralUnit>
      <InkMediaInformation>
        <OverlaidMedia>
          <OverlaidMediaRef href="#data"/>
        </OverlaidMedia>
        <Param name="From" value="foo@columbia.edu"/>
        <Param name="To" value="foo@mpeg.nist.gov"/>
      </InkMediaInformation>
    </InkSegment>
  </TemporalDecomposition>
</InkContent>
</MultimediaContent>
</Description>
</Mpeg7>

```

An example scenario to show the usage of the InkSegment DS is the following. At the end of the ink content creation, ink content is captured into an ink data file, and the metastrokes are described as a meta ink segment. At the same time, the scene graph, the media space, and the ordered group dataset masks locate the ink strokes stored in the ink document. A hand-writing recognition engine, described using HandWritingRecogInformation DS, can make use of the ink media and creation information and/or the stroke data to recognize words in ink strokes. If it succeeds, it reports the results using the HandWritingRecogResult DS in the InkSegment DS. As a result, a text keyword search can be used. However, if the hand-writing recognition engine fails to recognize the ink strokes, the time sequences of meta ink segments can be used for

matching. Although the masks are designed for the ink documents, they are general enough to be used for other media.

11.8.3 InkMediaInformation DS

11.8.3.1 Introduction

The InkMediaInformation DS describes specific media and creation information related to an ink segment. The InkMediaInformation DS describes the pen ink input device, the layout of the writing regions or fields in a form, the brush, the overlay of ink segment over other visual content, and the writing style. The InkMediaInformation DS applies to the InkSegment DS.

11.8.3.2 InkMediaInformation DS syntax

```
<!-- ##### -->
<!-- Definition of InkMediaInformation DS (11.8.3) -->
<!-- ##### -->

<!-- Definition of InkMediaInformation DS -->
<complexType name="InkMediaInformationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="InputDevice" minOccurs="0">
          <complexType>
            <sequence>
              <element name="Device" type="mpeg7:TermUseType"/>
            </sequence>
            <attribute name="xWidth" type="nonNegativeInteger"
              use="optional"/>
            <attribute name="yWidth" type="nonNegativeInteger"
              use="optional"/>
            <attribute name="resolutionX" type="nonNegativeInteger"
              use="optional"/>
            <attribute name="resolutionY" type="nonNegativeInteger"
              use="optional"/>
            <attribute name="resolutionZ" type="nonNegativeInteger"
              use="optional"/>
            <attribute name="resolutionP" type="integer"
              use="optional"/>
            <attribute name="resolutionS" type="integer"
              use="optional"/>
          </complexType>
        </element>
        <element name="WritingFieldLayout" minOccurs="0">
          <complexType>
            <sequence>
              <element name="RegionLocator"
                type="mpeg7:RegionLocatorType"/>
            </sequence>
            <attribute name="fieldIDList" use="optional">
              <simpleType>
                <list itemType="ID"/>
              </simpleType>
            </attribute>
          </complexType>
        </element>
        <element name="OverlaidMedia" minOccurs="0" maxOccurs="unbounded">
          <complexType>
```

```

<sequence>
  <choice>
    <element name="OverlaidMediaLocator"
      type="mpeg7:MediaLocatorType"/>
    <element name="OverlaidMediaRef"
      type="mpeg7:ReferenceType"/>
  </choice>
  <element name="MediaRegionLocator"
    type="mpeg7:RegionLocatorType" minOccurs="0"/>
</sequence>
<attribute name="fieldIDRef" type="IDREF" use="optional"/>
<attribute name="translationX" type="integer" use="optional"
  default="0"/>
<attribute name="translationY" type="integer" use="optional"
  default="0"/>
<attribute name="rotation" type="float" use="optional"
  default="0.0"/>
<attribute name="scaleX" type="float" use="optional"
  default="1.0"/>
<attribute name="scaleY" type="float" use="optional"
  default="1.0"/>
</complexType>
</element>
<element name="Brush" minOccurs="0" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element name="Name" type="mpeg7:TermUseType"
        minOccurs="0"/>
      <element name="Color" type="mpeg7:TermUseType"
        minOccurs="0"/>
      <element name="Shape" type="mpeg7:TermUseType"
        minOccurs="0"/>
      <element name="Size" type="nonNegativeInteger"
        minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<element name="Param" minOccurs="0" maxOccurs="unbounded">
  <complexType>
    <attribute name="name" type="string" use="optional"/>
    <attribute name="value" type="string" use="optional"/>
  </complexType>
</element>
<element name="Handedness" minOccurs="0">
  <simpleType>
    <restriction base="NMTOKEN">
      <enumeration value="left"/>
      <enumeration value="right"/>
      <enumeration value="others"/>
    </restriction>
  </simpleType>
</element>
<element name="Style" minOccurs="0">
  <simpleType>
    <union>
      <simpleType>
        <restriction base="NMTOKEN">
          <enumeration value="printed"/>
          <enumeration value="cursive"/>
          <enumeration value="drawing"/>
        </restriction>
      </simpleType>
    </union>
  </simpleType>
</element>

```

```

        <enumeration value="mixed"/>
    </restriction>
</simpleType>
<simpleType>
    <restriction base="mpeg7:termReferenceType"/>
</simpleType>
</union>
</simpleType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

```

11.8.3.3 InkMediaInformation DS semantics

Semantics of the InkMediaInformationType:

Name	Definition
InkMediaInformationType	Describes specific media and creation information related to an ink segment. InkMediaInformationType extends DType.
InputDevice	Describes the pen input device (optional).
Device	Identifies the input device with a name (optional).
xWidth	Indicates the width of the writing field's bounding box of the overall effective input surface of the device (application specified) in pixel units using the resolution of the input device (optional). This device parameter is independent of the display size.
yWidth	Indicates the height of the writing field's bounding box of the overall effective input surface of the device (application specified) in pixel units using the resolution of the input device (optional). This device parameter is independent of the display size.
resolutionX	Indicates the x resolution in points per millimeter of the input device (optional).
resolutionY	Indicates the y resolution in points per millimeter of the input device (optional).
resolutionZ	Indicates the z, i.e., altitude, resolution in points per millimeter of the input device (optional). It is preferred to use a linearized pressure in 1000 units per gram of force and calibrated with zero as the threshold of pen reaching the pad surface. Negative pressures can account for remaining non-linearities and hysteresis.
resolutionP	Indicates the pressure resolution in units per gram of the input device (optional). It is preferred to use a linearized pressure in 1000 units per gram of force and calibrated with zero as the threshold of pen reaching the pad surface. Negative pressures can account for remaining non-linearities and hysteresis.
resolutions	Indicates the sampling rate in points per seconds of the input device (optional).
WritingFieldLayout	Localizes the spatial layout of fragments of the ink segment (e.g., input block or guidelines) in the input device's writing field that guide the user in inputting the ink data and assigns an identifier to each region in the spatial layout (optional). The OverlaidMedia element uses this element.

Name	Definition
RegionLocator	Localizes spatial regions in the input device's writing field that guide the user in inputting the ink data. Each region will be assigned an identifier specified in the fieldIDList attribute.
fieldIDList	Identifies each region in the WritingFieldLayout element (optional). A writing field layout can be several writing fields localized by the RegionLocator element. This attribute assigns an identifier to each writing field in the order the RegionLocator element specifies them.
OverlaidMedia	Describes how the ink segment (or ink segment fragment) overlays over a sub-region of, or the entire visual content (optional). The OverlaidMedia element locates the specific regions of a visual content on which the specified ink segment or ink segment fragment overlays. It also defines the transformation of ink data points of the specified ink segment fragment before mapping. The mapping between the specified ink segment fragment and the regions of the visual content is a linear projection mapping.
OverlaidMediaLocator	Locates the visual content on which the ink segment or ink segment fragment overlays.
OverlaidMediaRef	Describes a reference to the description of a still region of the visual content on which the ink segment or ink segment fragment overlays.
MediaRegionLocator	Describes the spatial localization of the specific region within the visual content located by the OverlaidMediaLocator or OverlaidMediaRef element on which the entire ink segment or the ink segment fragment overlays (optional). If absent, the entire media located by OverlaidMediaLocator or OverlaidMediaRef element is overlaid. If the overlaid media is an ink segment, this element localizes a region of the writing field layout. All the ink data points within the specified region are overlaid.
fieldIDRef	References one of the regions in the writing field layout (optional). All the ink data points of the ink segment within the specified writing field overlay on the visual content specified by the OverlaidMedia element. If this attribute is absent, the entire ink segment overlays on the visual content.
translationX	Indicates the translation of ink data points (in the ink segment fragment specified by fieldIDRef) in the x direction with respect to the center of ink segment fragment (optional). The center of an ink segment is defined as the geometrical spatial coordinate center of all the ink data points of the ink segment.
translationY	Indicates the translation of ink data points (in the ink segment fragment specified by fieldIDRef) in the y direction with respect to the center of ink segment fragment (optional). Describes the transformation for the ink data points before linearly projecting the ink segment fragment onto the overlaid media region.
rotation	Indicates the counterclockwise angle rotation of the ink data points (in the ink segment fragment specified by fieldIDRef) with respect to the center of ink segment fragment (optional). Describes the transformation for the ink data points before linearly projecting the ink segment fragment onto the overlaid media region. This is a relative rotation; no initial angle is required.
scaleX	Indicates the scale factor of the ink data points in the ink segment fragment specified by fieldIDRef in the x direction (optional). Describes the transformation for the ink data points before linearly projecting the ink segment fragment onto the overlaid media region.

Name	Definition
scaleY	Indicates the scale factor of the ink data points (in the ink segment fragment specified by fieldIDRef) in the y direction (optional). Describes the transformation for the ink data points before linearly projecting the ink segment fragment onto the overlaid media region.
Brush	Describes a brush used to create the ink data (optional).
Name	Identifies the brush (optional).
Color	Describes the color of the brush (optional).
Shape	Describes the shape of the brush (optional).
Size	Describes the size of the brush (optional).
Param	Describes one parameter associated with an ink command (optional).
name	Describes the name of the parameter (optional).
value	Describes the value of the parameter (optional).
Handedness	Describes the writer's handedness (optional). The writer's handednesses are defined as follows: • <i>left</i> – The writer's handedness is left. • <i>right</i> – The writer's handedness is left. • <i>other</i> – The writer's handedness is other, i.e., a hand was not used to create the ink data.
Style	Describes style of ink data (optional). The styles of ink data are defined as follows: • <i>printer</i> – The style of the ink data is printer. • <i>cursive</i> – The style of the ink data is cursive. • <i>drawing</i> – The style of the ink data is drawing. • <i>mixed</i> – The style of the ink data is a mix of the above.

The **OverlaidMedia** element describes the region of visual content (the overlaid media) such as an image or an ink segment (e.g., form filling, highlight, command, and annotation) on which the ink segment or ink segment fragment overlays (the overlaying media). It describes the association and mapping between the overlaying and the overlaid media.

Note: the ink fragment segments specified in **WritingFieldLayout** element are defined in the coordinate space of the writing surface of the particular ink input device but the regions specified in the **OverlaidMedia** element are defined in the coordinate space of the overlaid media. Also, the orientation and scale of ink data points might not be aligned with the overlaid media. In the case of a meta ink segment as the overlaying media, this element identifies the media and region position that the metastroke operates on. In order to overlay ink data correctly over the overlaid media it is also required to specify the mapping transformation between the ink data points and the overlaid media. The mapping consists of two operations. The first operation is the alignment of the ink data points with the overlaid media. The second operation is simply the linear projection of the regions of ink data on overlaid media. The vertices of the regions of the overlaying ink segments are mapped into the vertices of the regions of the overlaid media.

11.8.3.4 InkMediaInformation DS examples (informative)

The following example illustrates the use of the InkMediaInformation DS for describing the input device and handedness information for an ink segment.

```
<InkMediaInformation>
  <InputDevice resolutionS="133">
    <Device>
      <Name xml:lang="en">PenDevice</Name>
    </Device>
  </InputDevice>
  <Handedness>right</Handedness>
  <Style>mixed</Style>
</InkMediaInformation>
```

The following example illustrates the use of the InkSegment DS and the OverlaidMedia element in the InkMediaInformation DS, among others, for describing ink content overlaying on a text medical form as shown in Figure 47. An example of application scenario for these kind of descriptions is the automatic collection of patient medical records for which the medical paper form stays on top of the writing surface of the ink input device. When the patient fills in the information on the paper form with a electronic pen, the information is simultaneously available as ink content.

In this example, the content ink segment is composed of three ink segments, each representing one word in the ink content. The WritingFieldLayout element describes the three rectangular regions shown in Figure 47, the overlaying media. The OverlaidMedia element in the InkSegment DS uses each region to map each region of the ink segment with the overlaid media, in this case, the medical paper form. The WritingFieldLayout element is useful for a handwriting recognizer or any other application that processes the ink content. It also provides the references for the ink content grouping (and segmentation). These guidelines may be visible in the input device writing surface depending on the application.

Medical Record Form

Family Name

Bernard

First Name

Anna

Health Condition

good

Figure 47 — Example of ink overlaid on a text medical form

This example also illustrates the use of transformation and region mapping of the OverlaidMedia element for describing the overlaying of the ink content on the text medical paper form. The visual regions and the ink segment fragments are defined in two different coordinate systems but their sizes are proportional for the first two writing field layouts, such that a linearly mapping between two regions does not distort the shape of ink strokes. For the first two writing field layouts, the ink data points are also well aligned within the ink region, and there is no need for and ink data translation. For the last writing field layout, the width of ink region "HealthCondition" is only one third of the width of visual region but there is not data translation either. The linear mapping of the third ink segment fragment results in the distortion (in the width direction) of the ink data. Display applications may decide to scale down the width of the third ink segment mapping by one third to maintain the shape of ink strokes.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="InkContentType">
      <InkContent>
        <SpatialDecomposition>
          <InkSegment>
            <InkMediaInformation>
              <WritingFieldLayout fieldIDList="LastName">
                <RegionLocator>
                  <Polygon>
                    <Coords mpeg7:dim="8"> 15 75 0 -6 45 0 0 6
                  </Coords>
                </Polygon>
              </RegionLocator>
            </WritingFieldLayout>
            <OverlaidMedia fieldIDRef="LastName">
              <OverlaidMediaLocator>
                <MediaUri>http://www.mpeg7.org/demo.txt</MediaUri>
              </OverlaidMediaLocator>
              <MediaRegionLocator>
                <Polygon>
                  <Coords mpeg7:dim="8"> 5 25 0 -2 15 0 0 2 </Coords>
                </Polygon>
              </MediaRegionLocator>
            </OverlaidMedia>
          </InkMediaInformation>
          <SceneGraphMask>
            <SubGraphNum>0</SubGraphNum>
          </SceneGraphMask>
        </InkSegment>
        <InkSegment>
          <InkMediaInformation>
            <WritingFieldLayout fieldIDList="FirstName">
              <RegionLocator>
                <Polygon>
                  <Coords mpeg7:dim="8"> 15 150 -6 45 0 0 6 </Coords>
                </Polygon>
              </RegionLocator>
            </WritingFieldLayout>
            <OverlaidMedia fieldIDRef="FirstName">
              <OverlaidMediaLocator>
                <MediaUri>http://www.mpeg7.org/demo.txt</MediaUri>
              </OverlaidMediaLocator>
              <MediaRegionLocator>
                <Polygon>
                  <Coords mpeg7:dim="8"> 5 50 0 -2 15 0 0 2 </Coords>
                </Polygon>
              </MediaRegionLocator>
            </OverlaidMedia>
          </InkMediaInformation>
          <SceneGraphMask>
            <SubGraphNum>1</SubGraphNum>
          </SceneGraphMask>
        </InkSegment>
        <InkSegment>
          <InkMediaInformation>
            <WritingFieldLayout fieldIDList="HealthCondition">
              <RegionLocator>
                <Polygon>
                  <Coords mpeg7:dim="8"> 15 225 -6 45 0 0 6 </Coords>
                </Polygon>
              </RegionLocator>
            </WritingFieldLayout>
            <OverlaidMedia fieldIDRef="HealthCondition">
              <OverlaidMediaLocator>
                <MediaUri>http://www.mpeg7.org/demo.txt</MediaUri>
              </OverlaidMediaLocator>
              <MediaRegionLocator>
                <Polygon>
                  <Coords mpeg7:dim="8"> 5 225 0 -2 15 0 0 2 </Coords>
                </Polygon>
              </MediaRegionLocator>
            </OverlaidMedia>
          </InkMediaInformation>
          <SceneGraphMask>
            <SubGraphNum>2</SubGraphNum>
          </SceneGraphMask>
        </InkSegment>
      </SpatialDecomposition>
    </InkContent>
  </MultimediaContent>
</Description>
</Mpeg7>

```



```

        </RegionLocator>
      </WritingFieldLayout>
      <OverlaidMedia fieldIDRef="HealthCondition" scaleX="0.3333">
        <OverlaidMediaLocator>
          <MediaUri>http://www.mpeg7.org/demo.txt</MediaUri>
        </OverlaidMediaLocator>
        <MediaRegionLocator>
          <Polygon>
            <Coords mpeg7:dim="8"> 5 75 0 -2 15 0 0 2 </Coords>
          </Polygon>
        </MediaRegionLocator>
      </OverlaidMedia>
    </InkMediaInformation>
    <SceneGraphMask>
      <SubGraphNum>2</SubGraphNum>
    </SceneGraphMask>
  </InkSegment>
</SpatialDecomposition>
</InkContent>
</MultimediaContent>
</Description>
</Mpeg7>

```

11.8.4 HandWritingRecogInformation DS

11.8.4.1 Introduction

The HandWritingRecogInformation DS describes information about a hand-writing recognizer. The HandWritingRecogInformation DS describes the owner and ink lexicon of the recognizer, among others. The HandWritingRecogInformationr DS applies to the InkSegment DS.

11.8.4.2 HandWritingRecogInformation DS syntax

```

<!-- ##### -->
<!-- Definition of HandWritingRecogInformation DS (11.8.4) -->
<!-- ##### -->

<!-- Definition of HandWritingRecogInformation DS -->
<complexType name="HandWritingRecogInformationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Recognizer" minOccurs="0">
          <complexType>
            <sequence>
              <element name="Name" type="mpeg7:TermUseType" minOccurs="0"/>
              <element name="Owner" type="mpeg7:AgentType" minOccurs="0"/>
              <element name="Information" type="mpeg7:TextAnnotationType"
                minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
        <element name="InkLexicon" minOccurs="0">
          <complexType>
            <sequence>
              <element name="Name" type="mpeg7:TermUseType" minOccurs="0"/>
              <element name="Owner" type="mpeg7:AgentType" minOccurs="0"/>
            <choice>

```

```

<element name="Resource" type="mpeg7:MediaLocatorType"/>
<element name="Entry" minOccurs="1" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element name="Text" type="mpeg7:TermUseType"
        minOccurs="0"/>
      <element name="InkSegmentRef"
        type="mpeg7:ReferenceType"
        minOccurs="0"/>
    </sequence>
    <attribute name="index" type="ID" use="optional"/>
    <attribute name="frequency"
      type="mpeg7:zeroToOneType" use="optional"/>
  </complexType>
</element>
</choice>
</sequence>
</complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

```

11.8.4.3 HandWritingRecogInformation DS semantics

Semantics of the HandWritingRecogInformationType:

Name	Definition
HandWritingRecogInformationType	Describes information about a hand-writing recognizer. HandWritingRecogInformationType extends DSType.
Recognizer	Describes the hand-writing recognizer (optional).
Name	Identifies the recognizer (optional). Humans can function as recognizers.
Owner	Describes the agent who provided the recognizer (optional).
Information	Describes the nature and structure of the recognizer such as the number of free parameters, the number of training examples, and the mechanism used, among others (optional).
InkLexicon	Describes the ink lexicon used by the recognizer locating the lexicon data or describing each entry in the lexicon (optional).
Name	Identifies the lexicon (optional).
Owner	Describes the owner of the lexicon who published the lexicon (optional).
Resource	Locates the lexicon, which does not need to be in electronic format.
Entry	Describes an entry in the lexicon. An entry in the lexicon is composed of a set of ink strokes.

Name	Definition
Text	Indicates the text string corresponding to the lexicon entry (optional).
InkSegmentRef	Describes a reference to the description of the ink segment that corresponds to the lexicon entry (optional).
index	Identifies a lexicon entry (optional).
frequency	Indicates the probability of appearance of the lexicon entry (optional). The frequencies characterize the lexicon's distribution from which data samples were drawn at random. Therefore, the number of times a lexicon entry appears in the database should be approximately proportional to the frequencies.

11.8.4.4 HandWritingRecogInformation DS examples (informative)

The following example illustrates the use of the HandWritingRecogInformation DS for describing information about a hand-writing recognizer named "Super Recog 1", which uses a lexicon with 20 entries representing names of people.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="InkContentType">
      <InkContent>
        <HandWritingRecogInformation>
          <Recognizer>
            <Name>
              <Name> Super Recog 1 </Name>
            </Name>
          </Recognizer>
          <InkLexicon>
            <Entry index="item0">
              <Text>
                <Name> Ann </Name>
              </Text>
              <InkSegmentRef href="#Electro0"/>
            </Entry>
            <Entry index="item1">
              <Text>
                <Name> Juan </Name>
              </Text>
              <InkSegmentRef href="#Electro1"/>
            </Entry>
            <!-- more elements here -->
            <Entry index="item20">
              <Text>
                <Name> Beatriz </Name>
              </Text>
              <InkSegmentRef href="#Electro20"/>
            </Entry>
          </InkLexicon>
        </HandWritingRecogInformation>
      </InkContent>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

11.8.5 HandWritingRecogResult DS

11.8.5.1 Introduction

The HandWritingRecogResult DS describes hand-writing recognition results. The HandWritingRecogResult DS describes the overall quality and some possible scored results of the recognition process. The HandWritingRecogResult DS applies to the InkSegment DS.

11.8.5.2 HandWritingRecogResult DS syntax

```
<!-- ##### -->
<!-- Definition of HandWritingRecogResult DS (11.8.5) -->
<!-- ##### -->

<!-- Definition of HandWritingRecogResult DS -->
<complexType name="HandWritingRecogResultType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="HandWritingRecogInformationRef"
          type="mpeg7:ReferenceType" minOccurs="0"/>
        <element name="Quality" type="mpeg7:zeroToOneType" minOccurs="0"/>
        <element name="Result" minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <choice>
                <element name="Text" type="string"/>
                <element name="LexiconIndexRef"
                  type="mpeg7:ReferenceType"/>
              </choice>
            </sequence>
            <attribute name="score" type="mpeg7:zeroToOneType"
              use="optional"/>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.8.5.3 HandWritingRecogResult DS semantics

Semantics of the HandWritingRecogResultType:

Name	Definition
HandWritingRecogResultType	Describes hand-writing recognition results. HandWritingRecogResultType extends DSType.
HandWritingRecogInformationRef	References the description of the hand-writing recognizer used to generate these results (optional).
Quality	Indicates the quality of the recognition results with a float between 0 and 1, being 1 the highest quality. This is used to describe the quality of the report score of the recognizer.

Name	Definition
Result	Describes a result of the recognition process by either the recognized text string or lexicon entry with an optional score value.
Text	Indicates the recognized text string.
LexiconIndexRef	References the recognized lexicon entry. This is a tool that refers to the index attribute of the Entry element in the HandWritingRecogInformation DS.
score	Indicates the reported score for the recognized text with a float between 0 and 1 being 1 the highest score (optional).

11.8.5.4 HandWritingRecogResult DS examples (informative)

The following example illustrates the use of the HandWritingRecogResult DS for describing the hand-writing recognition results for the word "Ana" in the ink document shown in Figure 45. The recognizer returns two results one of which corresponds to a lexicon entry in the HandWritingRecogInformation DS example in 11.8.4.4.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="InkContentType">
      <InkContent>
        <HandWritingRecogResult>
          <Quality>0.6</Quality>
          <Result score="0.7">
            <Text> Ana </Text>
          </Result>
          <Result score="0.54">
            <LexiconIndexRef href="#item0"/>
          </Result>
        </HandWritingRecogResult>
      </InkContent>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

11.8.6 Ink segment decomposition tools

11.8.6.1 Introduction

The ink segment decomposition tools extend the base segment decomposition tools. The ink segment decomposition tools describe the decomposition of an ink segment into one or more sub-segments in space and time.

11.8.6.2 Ink segment decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of InkSegment decomposition DSs (11.8.6) -->
<!-- ##### -->

<!-- Definition of InkSegmentSpatialDecomposition DS -->
<complexType name="InkSegmentSpatialDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatialSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
```

```

        <element name="InkSegment" type="mpeg7:InkSegmentType"/>
        <element name="InkSegmentRef" type="mpeg7:ReferenceType"/>
    </choice>
</extension>
</complexContent>
</complexType>

<!-- Definition of InkSegmentTemporalDecomposition DS -->
<complexType name="InkSegmentTemporalDecompositionType">
    <complexContent>
        <extension base="mpeg7:TemporalSegmentDecompositionType">
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="InkSegment" type="mpeg7:InkSegmentType"/>
                <element name="InkSegmentRef" type="mpeg7:ReferenceType"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

```

11.8.6.3 Ink segment decomposition tools syntax

Semantics of the InkSegmentSpatialDecompositionType:

Name	Definition
InkSegmentSpatialDecompositionType	Describes a spatial decomposition of an ink segment into one or more ink segments. InkSegmentSpatialDecompositionType extends SpatialSegmentDecompositionType.
InkSegment	Describes an ink segment that results from the spatial decomposition.
InkSegmentRef	Describes a reference to the description of an ink segment that results from the spatial decomposition.

Semantics of the InkSegmentTemporalDecompositionType:

Name	Definition
InkSegmentTemporalDecompositionType	Describes a temporal decomposition of an ink segment into one or more ink segments. InkSegmentTemporalDecompositionType extends TemporalSegmentDecompositionType.
InkSegment	Describes an ink segment that results from the temporal decomposition.
InkSegmentRef	Describes a reference to the description of an ink segment that results from the temporal decomposition.

11.9 Video editing segment description tools

11.9.1 Introduction

This subclause specifies tools for describing segments of visual content created by a video editing work, including attributes and structural decompositions of these segments. The following table summarizes the tools specified in this subclause. Figure 48 shows the organization of the tools for describing video editing segments.

Table 24 — Overview of the analytic edited video segment description tools

<i>Tool</i>	<i>Functionality</i>
AnalyticEditedVideo Segment DS	This tool is an abstract type that describes an edited video segment from an analytic viewpoint, which means that the description is made a posteriori automatically or manually based on the final video content. This tool extends the VideoSegment DS.
Analytic edited video segment decomposition tools	These tools describe decompositions of analytic edited video segments in space and time. These tools extend the base segment decomposition tools. The analytic edited video segment decomposition tools are the following: AnalyticEditedVideoSegmentSpatioTemporalDecomposition DS: describes the video editing decomposition of an analytic edited video segment in space and time.
EditedVideo DS	The EditedVideo DS describes a final video program that is the result of an editing work. This tool extends the AnalyticEditedVideoSegment DS.
Edited video decomposition tools	These tools describe decompositions of edited video segments in time. These tools extend the base segment decomposition tools. The edited video segment decomposition tools are the following: EditedVideoEditingTemporalDecomposition DS: describes the video editing decomposition of an edited video in time.
AnalyticClip DS	This tool is an abstract type that describes an intermediate edited video segment generated and assembled during the video editing process from the analytic viewpoint. Analytic clips are connected to other analytic clips by analytic transitions in the final video program. This tool extends the AnalyticEditedVideoSegment DS.
Shot DS	This tool describes a video segment that is delimited by transitions that affect the whole video screen (global transitions). This tool extends the AnalyticClip DS.
Shot decomposition tools	These tools describe decompositions of shots in time. These tools extend the base segment decomposition tools. The shot segment decomposition tools are the following: ShotEditingTemporalDecomposition DS: describes the video editing decomposition of a shot in time.
CompositionShot DS	This tool describes a video segment that is delimited by transitions caused by edition areas appearing on or disappearing from the video screen (composition transitions). This tool extends the AnalyticClip DS.
Composition shot decomposition tools	These tools describe decompositions of shots in time. These tools extend the base segment decomposition tools. The composition shot segment decomposition tools are the following:

Tool	Functionality
	CompositionShotEditingTemporalDecomposition DS: describes the video editing decomposition of a composition intra-composition shot in time.
IntraCompositionShot DS	This tool describes a video segment that is delimited by transitions that affect whole edition areas (internal transitions). This tool extends the AnalyticClip DS.
Intra-composition shot decomposition tools	These tools describe decompositions of intra-composition shots in time. These tools extend the base segment decomposition tools. The intra-composition shot segment decomposition tools are the following: IntraCompositionShotEditingTemporalDecomposition DS: describes the video editing decomposition of an intra-composition in time.
AnalyticTransition DS	This tool is an abstract type that describes a transition between two analytic clips in the final video program. This tool extends the AnalyticEditedVideoSegment DS.
GlobalTransition DS	This tool describes a transition that affects the whole video screen. This tool extends the AnalyticTransition DS.
CompositionTransition DS	This tool describes a transition caused by edition areas appearing on or disappearing from the video screen. This tool extends the AnalyticTransition DS.
InternalTransition DS	This tool describes a transition on that affects a whole edition area. This tool extends the AnalyticTransition DS.
EditedMovingRegion DS	This tool describes a moving region that corresponds to an edition area that contains a rush or a synthetic clip in the video content. This tool extends the MovingRegion DS.

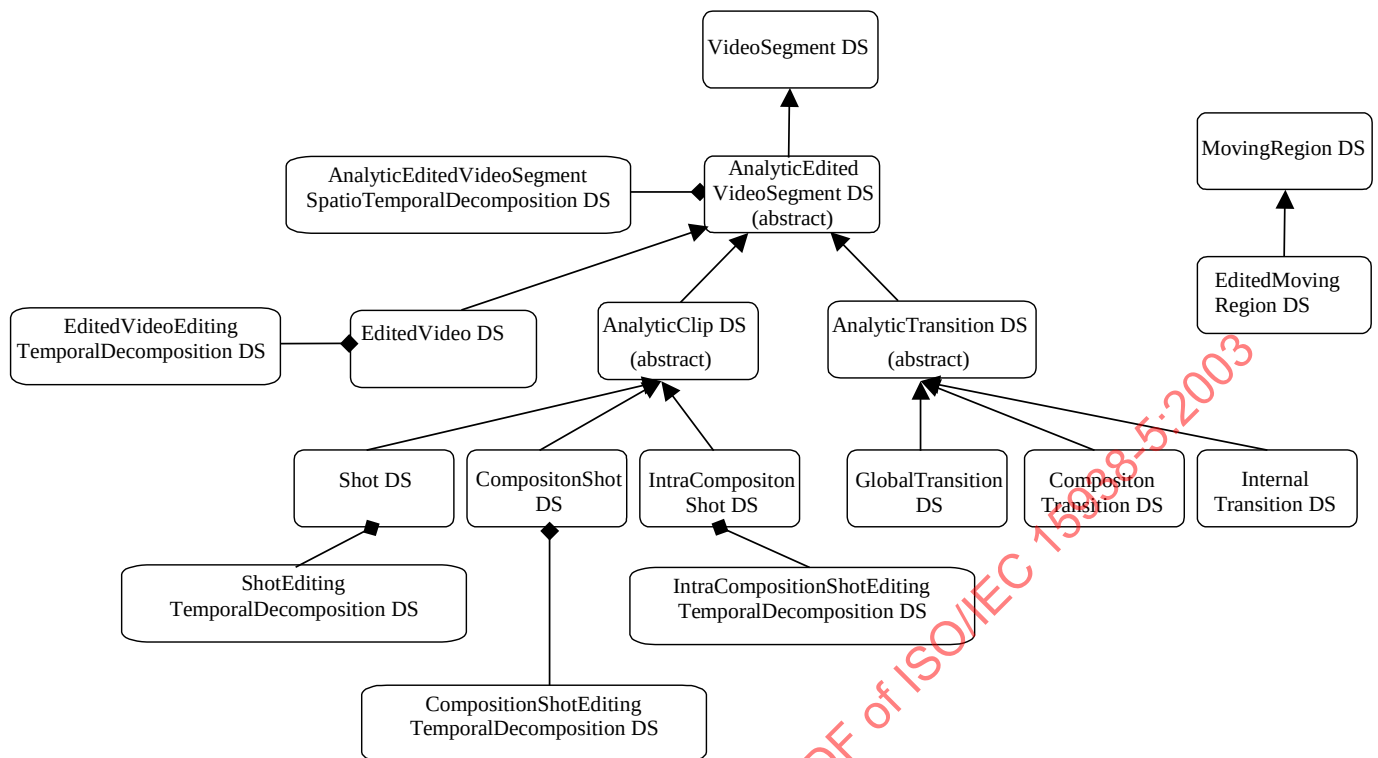


Figure 48 — Overview of the tools for describing video editing segments

11.9.2 AnalyticEditedVideoSegment DS

11.9.2.1 Introduction

The `AnalyticEditedVideoSegment` DS extends the `VideoSegment` DS. The `AnalyticEditedVideoSegment` DS is an abstract type that describes an edited video segment from an analytic viewpoint, which means that the description is made a posteriori automatically or manually based on the final video content. Examples of analytic edited video segments are a video shot and a video transition. The `AnalyticEditedVideoSegment` DS describes spatio-temporal decomposition from video editing viewpoint, location and editing level reliability, and type of the analytic edited video segment, in addition to the properties by the `VideoSegment` DS.

11.9.2.2 AnalyticEditedVideoSegment DS syntax

```
<!-- ##### -->
<!-- Definition of the AnalyticEditedVideoSegment DS (11.9.2) -->
<!-- ##### -->

<!-- Definition of AnalyticEditedVideoSegment DS -->
<complexType name="AnalyticEditedVideoSegmentType" abstract="true">
  <complexContent>
    <extension base="mpeg7:VideoSegmentType">
      <sequence>
        <element name="AnalyticEditionAreaDecomposition"
          type="mpeg7:AnalyticEditedVideoSegmentSpatioTemporalDecompositionType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="locationReliability" type="mpeg7:zeroToOneType"/>
    </complexContent>
  </complexType>

```

```

        use="optional"/>
    <attribute name="editingLevelReliability" type="mpeg7:zeroToOneType"
        use="optional"/>
    <attribute name="type" use="optional">
        <simpleType>
            <restriction base="NMTOKEN">
                <enumeration value="simple"/>
                <enumeration value="composite"/>
            </restriction>
        </simpleType>
    </attribute>
</extension>
</complexContent>
</complexType>

```

11.9.2.3 AnalyticEditedVideoSegment DS semantics

Semantics of the AnalyticEditedVideoSegmentType:

Name	Definition
AnalyticEditedVideoSegmentType	This type is an abstract type that describes an edited video segment from an analytic viewpoint, which means that the description is made a posteriori based on the final video content. Examples of analytic edited video segments are a shot and a global transition, among others. AnalyticEditedVideoSegmentType extends VideoSegmentType.
AnalyticEditionAreaDecomposition	Describes an spatio-temporal decomposition of the analytic edited video segment into one or more sub-segments that correspond to edition areas in the video segment. The edition areas describe video compositions. Composition transitions may introduce or move edition areas to and from the video screen, respectively. Internal transition may change the video content of edition areas.
locationReliability	Indicates the reliability of the temporal localization (media time information) of the analytic edited video segment (optional).
editingLevelReliability	Indicates the reliability of the classification of the analytic edited video segment into one of the editing levels (optional). If the analytic video segment is an analytic clip this attribute indicates the reliability of its description as a shot, a composition shot or an intra-composition shot. If the analytic video segment is an analytic transition this attribute indicates the reliability of its description as a global transition, a composition transition or an internal transition.
type	Indicates the type of the analytic edited video segment (optional). The types of analytic edited video segments are defined as follows: • <i>simple</i> - The analytic edited video segment is simple. The whole video screen in a simple analytic edited video segment is one edition area. • <i>composite</i> - The analytic edited video segment is composite. There are several local edition areas in the video screen of a composite analytic edited video segment. The AnalyticEditionAreaDecomposition element may describe the decomposition of the analytic edited video segment into edition areas.

11.9.2.4 Analytic edited video segments

11.9.2.4.1 Types

Video editing work consist in assembling and composing video clips. A clip is a video segment. In order, to compose video programs a set of edition regions or composition components are defined by giving its shape, size, position, motion and the way in which they are included on the frame space. The way to introduce a region edition is described by a composition transition. Once the region edition is defined, the editor identifies the set of clips that are placed in each edition region and determines how they shall be assembled.

An analytic description of video editing work describes the hierarchical structure of video clips and transitions generated during the editing process. In the process, clips corresponding shots, composition shots, and intra-composition shots are created. The definition of a "clip," "simple clip" and "composite clip" from analytic and synthetic viewpoint are given as follows:

- **Clip:** A clip is a video sequence. A clip is assembled to other clips by transitions. From an analytic viewpoint, a clip (called an *analytic clip*) is a video segment created by a video editing work. From a synthetic viewpoint, a clip (called a *synthetic clip*) is a piece of video that is assembled to other clips and placed into an edition area. A synthetic clip is frequently called "rush" in the video edition domain.
- **Simple clip:** Considering that a video frame is the smallest temporal unit of video content, a "simple clip" can be defined as a video frame sequence before being assembled with other synthetic clips to build a shot, a composition shot or an intra-composition shot (these are all analytic clips).
- **Composite clip:** A composite clip combines several synthetic clips in the same frame (see clause 8). Each synthetic clip is placed in one edition area (see Figure 49). One example is the superimposition of an anchor's name in TV news. There can be several composite clips in the same shot. An analytic clip may be defined as a composite clip even if the analytic edition area decomposition is not included in it.

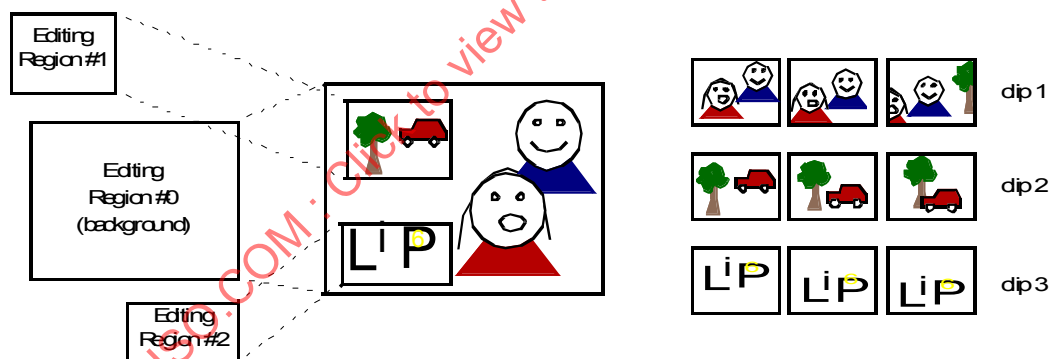


Figure 49 — Illustration of a composition transition. The central image is a frame of a composite clip. The clips shown on right are mapped on the edition regions shown on the left side

11.9.2.4.2 Editing levels

The analytic edited video segments are classified in two categories:

- Shots, composition shots and intra-composition shots are called analytic clips.
- Global transition, composition transition and internal transition are called analytic transitions.

They can be considered to be "syntactic elements" of a "video document". Analytic edited video segments are equivalent to "sentences" and the transitions to "punctuation" and "conjunctions".

The description of a video editing work corresponds to a hierarchical structure of three levels or more levels. The following table shows which are the corresponding levels regarding to the level of the analytic edited video segments.

Table 25 — Overview of the levels of analytic edited video segments

Editing Level	Analytic Clip	Analytic Transition
First level	Shot	Global transition
Second level	Composition shot	Composition transition
Third and higher levels	Intra-composition shot	Internal transition

A video document is made of a sequence of one or more shots and global transitions. A shot is made of a sequence of one or several composition shots and global transitions. A composition shot is made of a sequence of one or several intra-composition shots and internal transitions. An intra-composition shot is made of a sequence of one or more internal transitions and intra-composition shots, which are made, recursively, of one or more intra-composition shots and internal transitions. The analytic clips on a given level are linked by a transition of the same level. For example, shots are linked by global transitions.

11.9.3 Analytic edited video segment decomposition tools

11.9.3.1 Introduction

The analytic edited video segment decomposition tools extend the base segment spatio-temporal decomposition tools. The analytic edited video segment decomposition tools describe the video editing decomposition of an analytic edited video segment into one or more sub-segments in space and time. The sub-segments resulting from the decomposition correspond to edition areas in the analytic edited video segment.

11.9.3.2 Analytic video editing decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of the AnalyticEditedVideoSegment Decompostion DSs (11.9.3) -->
<!-- ##### -->

<!-- Definition of AnalyticEditedVideoSegmentSpatioTemporalDecomposition DS -->
<complexType name="AnalyticEditedVideoSegmentSpatioTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:SpatioTemporalSegmentDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="EditedMovingRegion"
          type="mpeg7:EditedMovingRegionType"/>
        <element name="EditedMovingRegionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

11.9.3.3 Analytic video editing decomposition tools semantics

Semantics of the AnalyticEditedVideoSegmentSpatioTemporalDecompositionType:

Name	Definition
AnalyticEditedVideoSegmentSpatioTemporalDecompositionType	Describes a spatio-temporal decomposition of an analytic edited video segment into one or more edited moving regions that correspond to edition areas in the analytic edited video segment. AnalyticEditedVideoSegmentSpatioTemporalDecompositionType

	extends SpatioTemporalSegmentDecompositionType.
EditedMovingRegion	Describes an edited moving region that results from the spatio-temporal decomposition.
EditedMovingRegionRef	Describes a reference to the description of an edited moving region that results from the spatio-temporal decomposition.

11.9.4 EditedVideo DS

11.9.4.1 Introduction

The EditedVideo DS extends the AnalyticEditedVideoSegment DS. The EditedVideo DS describes a final video program that is the result of an editing work. An edited video is a sequence of shots assembled by global transitions. The EditedVideo DS describes spatio-temporal decomposition from video editing viewpoint, in addition to the properties by the AnalyticEditedVideoSegment DS such as spatio-temporal decompositions into edition areas.

11.9.4.2 EditedVideo DS syntax

```

<!-- ##### -->
<!-- Definition of EditedVideo DS (11.9.4) -->
<!-- ##### -->

<!-- Definition of EditedVideo DS -->
<complexType name="EditedVideoType">
  <complexContent>
    <extension base="mpeg7:AnalyticEditedVideoSegmentType">
      <sequence>
        <element name="AnalyticEditingTemporalDecomposition"
          type="mpeg7:EditedVideoEditingTemporalDecompositionType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

11.9.4.3 EditedVideo DS semantics

Semantics of the EditedVideoType:

Name	Definition
EditedVideoType	Describes a final video program that is the result of an editing work. EditedVideoType extends AnalyticEditedVideoSegmentType.
AnalyticEditingTemporalDecomposition	Describes a temporal decomposition of the segment into one or more sub-segments that correspond to shots or global transitions.

11.9.5 Edited video decomposition tools

11.9.5.1 Introduction

The edited video decomposition tools extend the base segment temporal decomposition tools. The edited video decomposition tools describe the video editing decomposition of an edited video into one or more sub-segments in time. The sub-segments resulting from the decomposition correspond to shots or global transitions.

11.9.5.1.1 Edited video decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of the EditedVideoEditing decomposition DSs (11.9.5) -->
<!-- ##### -->

<!-- Definition of EditedVideoEditingTemporalDecomposition DS -->
<complexType name="EditedVideoEditingTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:TemporalSegmentDecompositionType">
      <sequence minOccurs="1" maxOccurs="unbounded">
        <choice>
          <choice>
            <element name="Shot" type="mpeg7:ShotType"/>
            <element name="ShotRef" type="mpeg7:ReferenceType"/>
          </choice>
          <choice>
            <element name="GlobalTransition" type="mpeg7:GlobalTransitionType"/>
            <element name="GlobalTransitionRef" type="mpeg7:ReferenceType"/>
          </choice>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.9.5.1.2 Edited video decomposition tools semantics

Semantics of the EditedVideoEditingTemporalDecompositionType:

Name	Definition
EditedVideoEditingTemporalDecompositionType	Describes a temporal decomposition of an edited video into one or more shots and global transitions. EditedVideoEditingTemporalDecompositionType extends TemporalSegmentDecompositionType. In an EditedVideoEditingTemporalDecomposition DS description, the Shot DS and GlobalTransition DS descriptions shall be ordered in time. Furthermore, a Shot DS description may be preceded and followed by a GlobalTransition DS description, but not by another Shot DS description.
Shot	Describes a shot that results from the temporal decomposition.
ShotRef	Describes a reference to the description of a shot that results from the temporal decomposition.

Name	Definition
GlobalTransition	Describes a global transition that results from the temporal decomposition.
GlobalTransitionRef	Describes a reference to the description of a global transition that results from the temporal decomposition.

11.9.6 AnalyticClip DS

11.9.6.1 Introduction

The AnalyticClip DS extends the AnalyticEditedVideoSegment DS. It is an abstract type from which specialized analytic clip DSs are derived: the Shot DS, the CompositionShot DS, and the IntraCompositionShot DS; which describe analytic clips at different video editing levels. The AnalyticClip DS describes an intermediate edited video segment generated and assembled during the video editing process from the analytic viewpoint. Analytic clips are connected to other analytic clips by analytic transitions in the final video program.

11.9.6.2 AnalyticClip DS syntax

```
<!-- ##### -->
<!-- Definition of AnalyticClip DS (11.9.6) -->
<!-- ##### -->

<!-- Definition of AnalyticClip DS -->
<complexType name="AnalyticClipType" abstract="true">
  <complexContent>
    <extension base="mpeg7:AnalyticEditedVideoSegmentType"/>
  </complexContent>
</complexType>
```

11.9.6.3 AnalyticClip DS semantics

Semantics of the AnalyticClipType:

Name	Definition
AnalyticClipType	This type is an abstract type from which the specialized analytic clip DSs are derived. It describes an intermediate edited video segment generated and assembled during the video editing process from the analytic viewpoint. Intermediate analytic clips are connected to other intermediate analytic clips by analytic transitions in the final video program. AnalyticClipType extends AnalyticEditedVideoSegmentType.

11.9.7 Shot DS

11.9.7.1 Introduction

A shot is a sequence of frames that forms a temporal unit in the video content. Its temporal boundaries are defined by the presence of transitions that affect the whole content (all the edition areas and their content) on the video frame: these transitions are called global transitions. Shots are created from simple and composite clips. Shots are the last video segments assembled during the video editing process. Final edited video programs are finally assembled by global transitions and constitute the shots

The Shot DS describes a video segment that is delimited by transitions that affect the whole video screen (global transitions). The Shot DS extends the AnalyticClip DS. Shots are the analytic clips at the first editing level and may be sequences of compositions shots assembled with composition transitions. The Shot DS describes temporal decomposition from the video editing viewpoint, in addition to the other properties by the AnalyticClip DS such as spatio-temporal decompositions into edition areas.

11.9.7.2 Shot DS syntax

```
<!-- ##### -->
<!-- Definition of Shot DS (11.9.7) -->
<!-- ##### -->

<!-- Definition of Shot DS -->
<complexType name="ShotType">
  <complexContent>
    <extension base="mpeg7:AnalyticClipType">
      <sequence>
        <element name="AnalyticEditingTemporalDecomposition"
          type="mpeg7:ShotEditingTemporalDecompositionType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```


11.9.7.3 Shot DS semantics

Semantics of the ShotType:

Name	Definition
ShotType	Describes a video segment that is delimited by transitions that affect the whole video screen (global transitions). Shots are the analytic clips at the first editing level and may be sequences of compositions shots assembled with composition transitions. The start and end of a video program are considered special cases of global transitions. ShotType extends AnalyticClipType.
AnalyticEditingTemporalDecomposition	Describes a temporal decomposition of the shot into one or more sub-segments that correspond to composition shots or composition transitions.

11.9.8 Shot decomposition tools

11.9.8.1 Introduction

The shot decomposition tools extend the base segment temporal decomposition tools. The shot decomposition tools describe the video editing decomposition of a shot into one or more sub-segments in time. The sub-segments resulting from the decomposition correspond to composition shots or composition transitions.

11.9.8.2 Shot decomposition tools syntax

```

<!-- ##### -->
<!-- Definition of the ShotEditing Decomposition DSs (11.9.8) -->
<!-- ##### -->
<!-- Definition of the ShotEditingTemporalDecomposition DS -->

<complexType name="ShotEditingTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:TemporalSegmentDecompositionType">
      <sequence minOccurs="1" maxOccurs="unbounded">
        <choice>
          <choice>
            <element name="CompositionShot" type="mpeg7:CompositionShotType"/>
            <element name="CompositionShotRef" type="mpeg7:ReferenceType"/>
          </choice>
          <choice>
            <element name="CompositionTransition"
              type="mpeg7:CompositionTransitionType"/>
            <element name="CompositionTransitionRef" type="mpeg7:ReferenceType"/>
          </choice>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

11.9.8.3 Shot decomposition tools semantics

Semantics of the ShotEditingTemporalDecompositionType:

Name	Definition
ShotEditingTemporalDecompositionType	Describes a temporal decomposition of a shot into one or more composition shots and composition transitions. ShotEditingTemporalDecompositionType extends TemporalSegmentDecompositionType. In an ShotEditingTemporalDecomposition DS description, the CompositionShot DS and CompositionTransition DS descriptions shall be ordered in time. Furthermore, a CompositionShot DS description can be preceded and followed by a CompositionTransition DS description, but not by another CompositionShot DS description.
CompositionShot	Describes a composition shot that results from the temporal decomposition.
CompositionShotRef	Describes a reference to the description of a composition shot that results from the temporal decomposition.
CompositionTransition	Describes a composition transition that results from the temporal decomposition.
CompositionTransitionRef	Describes a reference to the description of a composition transition that results from the temporal decomposition.

11.9.9 CompositionShot DS

11.9.9.1 Introduction

Composition shots are temporal sub-segments of a shot separated by transitions called composition transitions. Composition shots are the analytic clips at the second level in hierarchical temporal structure of video documents created by editing work. They are sub-segments of a shot. In a shot an edition region can be introduced or removed by a composition transition. This temporal event split the shot in a new video segment called composition shot.

The CompositionShot DS describes a video segment that is delimited by transitions caused by edition areas appearing on or disappearing from the video screen (composition transitions). The CompositionShot DS extends the AnalyticClip DS. The Composition shots are analytic clips at the second editing level and may be sequences of intra-composition shots assembled with internal transitions. The CompositionShot DS describes temporal decomposition from the video editing viewpoint, in addition to the other properties by the AnalyticClip DS such as spatio-temporal decompositions into edition areas.

11.9.9.2 CompositionShot DS syntax

```
<!-- ##### -->
<!-- Definition of CompositionShot DS (11.9.9) -->
<!-- ##### -->

<!-- Definition of CompositionShot DS -->
<complexType name="CompositionShotType">
```

```

<complexContent>
  <extension base="mpeg7:AnalyticClipType">
    <sequence>
      <element name="AnalyticEditingTemporalDecomposition"
        type="mpeg7:CompositionShotEditingTemporalDecompositionType"
        minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </extension>
</complexContent>
</complexType>

```

11.9.9.3 CompositionShot DS semantics

Semantics of the CompositionShotType:

Name	Definition
CompositionShotType	Describes a video segment that is delimited by transitions caused by edition areas appearing on or disappearing from the video screen (composition transitions). Composition shots are analytic clips at the second editing level and may be sequences of intra-composition shots assembled with internal transitions. Transitions that affect the whole video screen (global transitions) are considered special cases of composition transitions but they can only be one of the boundaries of composition shot. CompositionShotType extends AnalyticClipType.
AnalyticEditingTemporalDecomposition	Describes a temporal decomposition of the composition shot into one or more sub-segments that correspond to intra-composition shots or internal transitions.

11.9.10 Composition shot decomposition tools

11.9.10.1 Introduction

The composition shot decomposition tools extend the base segment temporal decomposition tools. The composition shot decomposition tools describe the video editing decomposition of a composition shot into one or more sub-segments in time. The sub-segments resulting from the decomposition correspond intra-composition shots or internal transitions.

11.9.10.2 Composition shot decomposition tools syntax

```

<!-- ##### -->
<!-- Definition of the CompositionShotEditing decomposition DSs (11.9.10) -->
<!-- ##### -->
<complexType name="CompositionShotEditingTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:TemporalSegmentDecompositionType">
      <sequence minOccurs="1" maxOccurs="unbounded">
        <choice>
          <choice>
            <element name="IntraCompositionShot"
              type="mpeg7:IntraCompositionShotType"/>
            <element name="IntraCompositionShotRef" type="mpeg7:ReferenceType"/>

```

```

        </choice>
        <choice>
            <element name="InternalTransition"
type="mpeg7:InternalTransitionType"/>
            <element name="InternalTransitionRef" type="mpeg7:ReferenceType"/>
        </choice>
    </choice>
</sequence>
</extension>
</complexContent>
</complexType>

```

11.9.10.3 Composition shot decomposition tools semantics

Semantics of the CompositionShotEditingTemporalDecompositionType:

Name	Definition
CompositionShotEditingTemporalDecompositionType	Describes a temporal decomposition of a composition shot into one or more intra-composition shots and internal transitions. CompositionShotEditingTemporalDecompositionType extends TemporalSegmentDecompositionType. In a CompositionShotEditingTemporalDecomposition, the IntraCompositionShot and InternalTransition shall be ordered in time. Furthermore, an IntraCompositionShot DS description can be preceded and followed by a InternalTransition DS description, but not by another IntraCompositionShot DS description.
IntraCompositionShot	Describes an intra-composition shot that results from the temporal decomposition.
IntraCompositionShotRef	Describes a reference to the description of an intra-composition shot that results from the temporal decomposition.
InternalTransition	Describes an internal transition that results from the temporal decomposition.
InternalTransitionRef	Describes a reference to the description of an internal transition that results from the temporal decomposition.

11.9.11 IntraCompositionShot DS

11.9.11.1 Introduction

Intra-composition shots are temporal sub-segments of a composition shot or another intra-composition shot separated by internal transitions. These transitions occur between two synthetic clips appearing in one of the regions of a "composite clip". These transitions modify the content of the composition shot or an intra-composition shot and split it into two new segments called intra-composition shots.

Intra-composition shots are analytic clips at the third and higher levels in the hierarchical temporal structure of video documents created by editing work. Intra-composition shots can be recursively split in other intra-composition shots. They are sub-segments of a composition shot or other intra-composition shot:

1. In a composition shot, the synthetic clip showed in one of the edition regions can be changed by another one by an internal transition. This temporal event split the composition shot in a new video segment called intra-composition shot. One example is shown in Figure 50.
2. In an intra-composition shot, an internal transition can modify the content of an sub-edition region within an edition area, or a new sub-edition region can be added or moved from an already existing edition area. These temporal events split the intra-composition shot in a new video segment called intra-composition-shot. This new intra-composition shot is in a lower level than its intra-composition shot parent.

One example of intra-composition shots in a composition shot is given in Figure 50. In the CompositionShot#2, there are two edition regions: the background and a superimposed circular region showing a doctor. At a given moment, an internal transition changes the doctor by a dog in the circular edition region. This internal transition split the CompositionShot#2 in a new temporal segment, an intra-composition shot: IntraCompositionShot#2.

Because an internal transition assembles intra-composition shots, the segment before the internal transition is also called intra-composition shot.

The IntraCompositionShot DS describes a video segment that is delimited by transitions that affect whole edition areas (internal transitions). The IntraCompositionShot DS extends the AnalyticClip DS. Intra-composition shots are the analytic clips at the third and higher editing levels and may recursively be sequences of intra-compositions shots assembled with internal transitions. The IntraCompositionShot DS describes temporal decomposition from the video editing viewpoint, in addition to the other properties by the AnalyticClip DS such as spatio-temporal decompositions into edition areas.

11.9.11.2 IntraCompositionShot DS syntax

```
<!-- ##### -->
<!-- Definition of IntraCompositionShot DS (11.9.11) -->
<!-- ##### -->

<!-- Definition of IntraCompositionShot DS -->
<complexType name="IntraCompositionShotType">
  <complexContent>
    <extension base="mpeg7:AnalyticClipType">
      <sequence>
        <element name="AnalyticEditingTemporalDecomposition"

          type="mpeg7:IntraCompositionShotEditingTemporalDecompositionType"
            minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.9.11.3 IntraCompositionShot DS semantics

Semantics of the IntraCompositionShotType:

Name	Definition
IntraCompositionShotType	Describes a video segment that is delimited by transitions that affect whole edition areas (internal transitions). Intra-composition shots are the analytic clips at the third and higher editing levels and may recursively be sequences of intra-compositions shots assembled with internal transitions. Transitions that affect the whole video screen (global transitions) and transitions caused by edition areas appearing on or disappearing from the video screen (composition transitions) are considered special cases of internal transitions but they can only be one of the boundaries of an intra composition shot. IntraCompositionShotType extends AnalyticClipType.

AnalyticEditingTemporalDecomposition	Describes a temporal decomposition of the intra-composition shot into one or more sub-segments that correspond to intra-composition shots or internal transitions.

11.9.12 Intra-composition shot decomposition tools

11.9.12.1 Introduction

The Intra-composition shot decomposition tools extend the base segment temporal decomposition tools. The intra-composition shot decomposition tools describe the video editing decomposition of an intra-composition shot into one or more sub-segments in time. The sub-segments resulting from the decomposition correspond to intra-composition shots or internal transitions.

11.9.12.2 Intra-composition shot decomposition tools syntax

```
<!-- ##### -->
<!-- Definition of the IntraCompositionShotEditing decomposition DSs (11.9.12) -->
<!-- ##### -->
<!-- Definition of IntraCompositionShotEditingTemporalDecomposition DS -->

<complexType name="IntraCompositionShotEditingTemporalDecompositionType">
  <complexContent>
    <extension base="mpeg7:TemporalSegmentDecompositionType">
      <sequence minOccurs="1" maxOccurs="unbounded">
        <choice>
          <choice>
            <element name="IntraCompositionShot"
              type="mpeg7:IntraCompositionShotType"/>
            <element name="IntraCompositionShotRef" type="mpeg7:ReferenceType"/>
          </choice>
          <choice>
            <element name="InternalTransition"
              type="mpeg7:InternalTransitionType"/>
            <element name="InternalTransitionRef" type="mpeg7:ReferenceType"/>
          </choice>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.9.12.3 Intra-composition shot decomposition tools semantics

Semantics of the IntraCompositionShotEditingTemporalDecompositionType:

Name	Definition
IntraCompositionShotEditingTemporalDecompositionType	Describes a temporal decomposition of an intra-composition shot into one or more intra-composition shots and internal transitions. IntraCompositionShotEditingTemporalDecompositionType extends TemporalSegmentDecompositionType. In an IntraCompositionShotEditingTemporalDecomposition, the IntraCompositionShot and InternalTransition shall be ordered in time. Furthermore, an IntraCompositionShot DS

	description can be preceded and followed by a InternalTransition DS description, but not by another IntraCompositionShot DS description.
IntraCompositionShot	Describes an intra-composition shot that results from the temporal decomposition.
IntraCompositionShotRef	Describes a reference to the description of an intra-composition shot that results from the temporal decomposition.
InternalTransition	Describes an internal transition that results from the temporal decomposition.
InternalTransitionRef	Describes a reference to the description of an internal transition that results from the temporal decomposition.

11.9.13 AnalyticTransition DS

11.9.13.1 Introduction

The AnalyticTransition DS extends the AnalyticEditedVideoSegment DS. It is an abstract type from which specialized analytic transition DSs are derived: the GlobalTransition DS, the CompositionTransition DS, and the InternalTransition DS; which describe analytic transitions at different video editing levels. The AnalyticTransition DS describes a transition between two analytic clips in the final video program. Transitions are generated during the video editing process to connect shots, composition shots, or intra-composition shots. The AnalyticTransition DS describes evolution type and reliability, in addition to the other properties by the AnalyticEditedVideoSegment DS.

11.9.13.2 AnalyticTransition DS syntax

```
<!-- ##### -->
<!-- Definition of AnalyticTransition DS (11.9.13) -->
<!-- ##### -->

<!-- Definition of AnalyticTransition DS -->
<complexType name="AnalyticTransitionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:AnalyticEditedVideoSegmentType">
      <sequence>
        <element name="EvolutionType" type="mpeg7:TermUseType" minOccurs="0"/>
        <element name="EditedMovingRegionRef" type="mpeg7:ReferenceType"
          minOccurs="0"/>
      </sequence>
      <attribute name="evolutionReliability" type="mpeg7:zeroToOneType"
        use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

11.9.13.3 AnalyticTransition DS semantics

Semantics of the AnalyticTransitionType:

Name	Definition
AnalyticTransitionType	This type is an abstract type from which the specialized analytic transition DSs are derived. It describes a transition between two analytic clips in the final video program. AnalyticTransitionType extends AnalyticEditedVideoSegmentType.
EvolutionType	Describes the type of the evolution of the analytic transition (optional). Examples of evolution types are "cut", "analog cut", and "gradual transitions". For gradual transitions, examples of evolution types are "fade in", "fade out" and "cross-dissolve", among others.
EditedMovingRegionRef	Describes a reference to the description of the edited moving region (edition area) that is affected, introduced or deleted in the analytic transition (optional).
evolutionReliability	Indicates the reliability of the evolution information of the analytic transition (optional). This attribute specifies the reliability of the value of the EvolutionType element, if provided.

11.9.14 GlobalTransition DS

11.9.14.1 Introduction

A global transition is a visual effect that covers the whole video frame. This transition supposes that the whole shot (including the "sub-clips" it may be composed of) on the video frame before the effect disappears, and that a new one which was not visible until that moment, appears after the transition. Figure 50 shows an example of an global transition.

The GlobalTransition DS describes a transition that affects the whole video screen. The GlobalTransition DS extends the AnalyticTransition DS. Global transitions are the analytic transitions at the first video editing level and connect shots in edited videos. The GlobalTransition DS describes properties of the global transition by the AnalyticTransition DS such as spatio-temporal decompositions into edition areas, evolution type and reliability.

11.9.14.2 GlobalTransition DS syntax

```
<!-- ##### -->
<!-- Definition of GlobalTransition DS (11.9.14) -->
<!-- ##### -->

<!-- Definition of GlobalTransition DS -->
<complexType name="GlobalTransitionType">
  <complexContent>
    <extension base="mpeg7:AnalyticTransitionType"/>
  </complexContent>
</complexType>
```


11.9.14.3 GlobalTransition DS semantics

Semantics of the GlobalTransitionType:

Name	Definition
GlobalTransitionType	Describes a transition that affects the whole video screen. Global transitions are the analytic transitions at the first video editing level and connect shots in edited videos. A global transition supposes that a shot disappears before the transition and that a shot that was not visible until that moment appears after the transition. A global transition can be one of the boundaries of a composition shot and an intra-composition shot. GlobalTransitionType extends AnalyticTransitionType.

11.9.15 CompositionTransition DS

11.9.15.1 Introduction

Composition transitions are caused by edition areas appearing on, or disappearing from the video frame. As opposed to global transitions, composition transitions only affect editions areas in the video frame. Figure 50 shows examples of composition transitions.

Composition transitions can perform one of the three following actions:

- **Create:** the composition transition introduces an edition area that creates a composite clip. It links a composition shot containing a single edition area and a composition shot combining several edition areas.
- **Delete:** the composition transition ends a composite clip. It links the end of a composition shot combining several edition areas and the beginning of a composition shot containing a single edition area.
- **Change:** this last type is related to transitions that add or remove edition areas between two composition shots combining several edition areas.

The type of the action in a transition depends on the type of the composition shots the transition connects:

Action	Number of edition areas in previous composition shot	Number of edition areas in next composition shot
Create	1	>1
Delete	>1	1
Change	>1	>1

One example of the first type of action is the superimposition of the anchor's name in a TV news. This text can be introduced gradually by a dissolve effect which is a composition transition.

Edition areas taking part in a composite clip, like texts, may move within the frame. This motion does not create a new composition shot. A composition transition occurs only when an edition area appears on, or disappears from the video frame.

The CompositionTransition DS describes a transition caused by edition areas appearing on or disappearing from the video screen. The CompositionTransition DS extends the AnalyticTransition DS. Composition transitions are analytic transitions at the second video editing level and connect composition shots in shots. The CompositionTransition DS describes composition type of the composition transition, in addition to other properties by the AnalyticTransition DS such as spatio-temporal decompositions into edition areas, evolution type and reliability.

11.9.15.2 CompositionTransition DS syntax

```

<!-- ##### -->
<!-- Definition of CompositionTransition DS (11.9.15) -->
<!-- ##### -->

<!-- Definition of CompositionTransition DS -->
<complexType name="CompositionTransitionType">
  <complexContent>
    <extension base="mpeg7:AnalyticTransitionType">
      <attribute name="compositionType" use="optional">
        <simpleType>
          <union>
            <simpleType>
              <restriction base="NMTOKEN">
                <enumeration value="create"/>
                <enumeration value="delete"/>
                <enumeration value="change"/>
              </restriction>
            </simpleType>
            <simpleType>
              <restriction base="mpeg7:termReferenceType"/>
            </simpleType>
          </union>
        </simpleType>
      </attribute>
    </extension>
  </complexContent>
</complexType>

```

11.9.15.3 CompositionTransition DS semantics

Semantics of the CompositionTransitionType:

Name	Definition
CompositionTransitionType	Describes a transition caused by edition areas appearing on or disappearing from the video screen. Composition transitions are the analytic transitions at the second video editing level and connect composition shots in shots. A composition transition can be one of the boundaries of an intra-composition shot. CompositionTransitionType extends AnalyticTransitionType.
compositionType	Indicates the type of composition of the composition transition in relation to the type of the action performed on the composition shot (optional). The types of composition are defined as follows: <i>create</i> - The composition introduces an edition area that creates a composite clip. In other words, the transition links a composition shot containing a single edition area and a composition shot combining several edition areas. <i>delete</i> - The composition ends a composite clip. In other words, the transition links the end of a composition shot combining several edition areas and the beginning of a composition shot containing a single edition area. <i>change</i> - The composition adds or removes edition areas between two composition shots combining several edition areas.

11.9.16 InternalTransition DS

11.9.16.1 Introduction

An internal transitions happens within composition shots and within intra-composition shots splitting these segments in intra-composition shots. An internal transition happens within a composition shot when there is a global transition in one of the edition areas of the video frame. This transition has all the characteristics of a global transition but it happens only in a sub-region of the video frame. It also differs from global transitions by the level of the analytic edited video segments connected by the transition (only intra-composition shots for the internal transition and shots for the global transition). Figure 50 shows an example of internal transitions. The first shot contains two different transitions. First, there is a composition transition of type "cut", where a superimposition of a new content occurs in a subpart of the frame. The second is an internal transition: the superimposed edition area is not affected by the effect. A transition of type "cut" introduces the second shot. An internal transition happens within in intra-composition shot when a new sub-edition region is added or moved from an existing edition region, or when there is a global transition within a sub-edition area of an existing edition region.

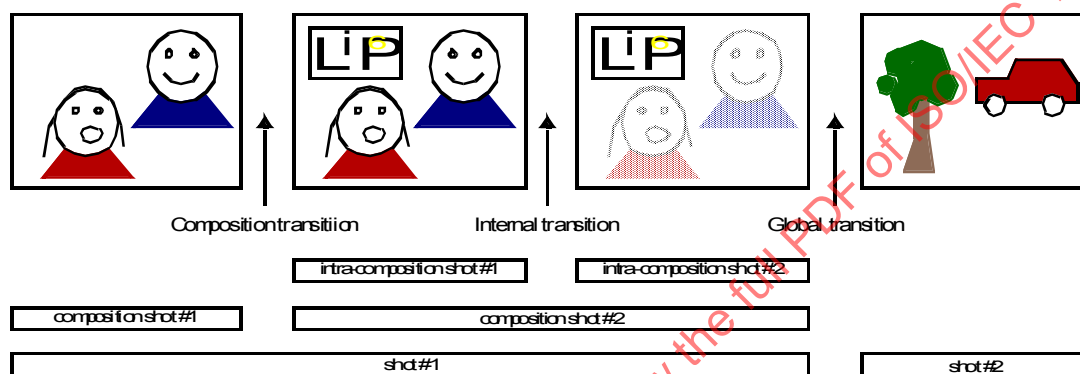


Figure 50 — Illustration of different types of transitions (composition transition, internal transition, global transition)

The InternalTransition DS describes a transition on that affects a whole edition area. The InternalTransition DS extends the AnalyticTransition DS. Internal transitions are the analytic transitions at the third and higher video editing levels and connect intra-composition shots in composition shots. The InternalTransition DS describes properties of the internal transition by the AnalyticTransition DS such as spatio-temporal decompositions into edition areas, evolution type and reliability.

11.9.16.2 InternalTransition DS syntax

```
<!-- ##### -->
<!-- Definition of InternalTransition DS (11.9.16) -->
<!-- ##### -->

<!-- Definition of InternalTransition DS -->
<complexType name="InternalTransitionType">
  <complexContent>
    <extension base="mpeg7:AnalyticTransitionType"/>
  </complexContent>
</complexType>
```

11.9.16.3 InternalTransition DS semantics

Semantics of the InternalTransitionType:

Name	Definition
InternalTransitionType	Describes a transition on that affects a whole edition area. Internal transitions are the analytic transitions at the third and higher video editing levels and connect intra-composition shots in composition shots. InternalTransitionType extends AnalyticTransitionType.

11.9.17 EditedMovingRegion DS**11.9.17.1 Introduction**

The EditedMovingRegion DS extends the MovingRegion DS. The EditedMovingRegion DS describes a moving region that corresponds to an edition area that contains a rush or a synthetic clip in the video content. Examples of edited moving regions are edition areas that contain logo or anchors' names.

11.9.17.2 EditedMovingRegion DS syntax

```
<!-- ##### -->
<!-- Definition the EditedMovingRegion DS (11.9.17) -->
<!-- ##### -->

<!-- Definition of EditedMovingRegion DS -->
<complexType name="EditedMovingRegionType">
  <complexContent>
    <extension base="mpeg7:MovingRegionType">
      <sequence>
        <element name="RegionDepthType" type="mpeg7:TermUseType"
          minOccurs="0"/>
        <element name="BoundaryType" type="mpeg7:TermUseType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

11.9.17.3 EditedMovingRegion DS semantics

Semantics of the EditedMovingRegionType:

Name	Definition
EditedMovingRegionType	Describes a moving region that corresponds to an edition area that contains a rush or a synthetic clip in the video content. Examples of edited moving regions are edition areas that contain logos or anchors' names. EditedMovingRegionType extends MovingRegionType.
RegionDepthType	Describes the relative distance of the edited moving region to the camera (optional). Examples of region depths are "background" and "foreground".
BoundaryType	Describes the type of boundary of the edited moving region (optional). Examples of boundary types are "transparent", "fuzzy", and "color".

11.9.18 Video editing segment description tools examples (informative)

11.9.18.1 Introduction

The following examples illustrate the use of the AnalyticEditedVideo DS and the video editing segment DSs for describing several video editing works. Descriptions of video editing works can be used for video indexing and analysis, evaluations video segmentation algorithms, and identification of editing patterns.

11.9.18.2 Description of a sequence of composition shots and internal transitions

The following example illustrates the use of the AnalyticClip DS and the AnalyticTransition DS for describing a video content that contains a single shot with two composition shots and an internal transition (see Figure 50).

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AnalyticEditedVideoType">
      <AnalyticEditedVideo xsi:type="EditedVideoType">
        <MediaLocator>
          <MediaUri>video.mpg</MediaUri>
        </MediaLocator>
        <!-- more elements here -->
        <AnalyticEditingTemporalDecomposition gap="false" overlap="false">
          <!-- Description of Eyexam -->
          <Shot id="ClipS_3" locationReliability="0.9"
            editingLevelReliability="0.5">
            <MediaTime>
              <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">35</MediaRelIncrTimePoint>
              <MediaIncrDuration
mediaTimeUnit="PT1N25F">413</MediaIncrDuration>
            </MediaTime>
            <AnalyticEditingTemporalDecomposition gap="false"
              overlap="false">
              <CompositionShot id="ClipCS_1" locationReliability="0.9"
                editingLevelReliability="0.5" type="simple">
                <MediaTime>
                  <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">35</MediaRelIncrTimePoint>
                  <MediaIncrDuration
mediaTimeUnit="PT1N25F">55</MediaIncrDuration>
                </MediaTime>
                </CompositionShot>
                <CompositionTransition id="CT_1" locationReliability="0.9"
                  editingLevelReliability="0.5" evolutionReliability="0.5">
                  <MediaTime>
                    <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">90</MediaRelIncrTimePoint>
                    <MediaIncrDuration
mediaTimeUnit="PT1N25F">10</MediaIncrDuration>
                  </MediaTime>
                  <EvolutionType href="...">
                    <Name>IrisOpen</Name>
                  </EvolutionType>
                </CompositionTransition>
                <CompositionShot id="ClipCS_2" locationReliability="0.9"
                  editingLevelReliability="0.5">
                  <MediaTime>
                    <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">100</MediaRelIncrTimePoint>
```

```

        <MediaIncrDuration
            mediaTimeUnit="PT1N25F">348</MediaIncrDuration>
        </MediaTime>
        <AnalyticEditingTemporalDecomposition gap="false"
            overlap="false">
            <IntraCompositionShot id="ClipICS_1"
                locationReliability="0.9"
                editingLevelReliability="0.5" type="simple">
                <MediaTime>
                    <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../../../../../MediaLocator[1]">100</MediaRelIncrTimePoint>
                    <MediaIncrDuration
                        mediaTimeUnit="PT1N25F">75</MediaIncrDuration>
                    </MediaTime>
                </IntraCompositionShot>
                <InternalTransition id="IT_1"
                    locationReliability="0.9"
                    editingLevelReliability="0.5"
                    evolutionReliability="0.5">
                    <MediaTime>
                        <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../../../../../MediaLocator[1]">175</MediaRelIncrTimePoint>
                        <MediaIncrDuration
                            mediaTimeUnit="PT1N25F">8</MediaIncrDuration>
                        </MediaTime>
                        <EvolutionType href="...">
                            <Name xml:lang="en">crossdissolve</Name>
                        </EvolutionType>
                    </InternalTransition>
                    <IntraCompositionShot id="ClipICS_2"
                        locationReliability="0.9"
                        editingLevelReliability="0.5" type="simple">
                        <MediaTime>
                            <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../../../../../MediaLocator[1]">183</MediaRelIncrTimePoint>
                            <MediaIncrDuration
                                mediaTimeUnit="PT1N25F">265</MediaIncrDuration>
                            </MediaTime>
                        </IntraCompositionShot>
                    </AnalyticEditingTemporalDecomposition>
                </CompositionShot>
            </AnalyticEditingTemporalDecomposition>
        </Shot>
    </AnalyticEditingTemporalDecomposition>
</AnalyticEditedVideo>
</MultimediaContent>
</Description>
</Mpeg7>

```

11.9.18.3 Description of a shot associated to a composite clip

The following examples illustrate the use of the AnalyticClip DS, the AnalyticTransition DS, and the EditedMovingRegion DS for describing a shot that is a composite clip. In the first example, there are no internal transitions: the analytic clip is a composite clip and a shot.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AnalyticEditedVideoType">
      <AnalyticEditedVideo xsi:type="EditedVideoType">
        <MediaLocator>
          <MediaUri>video.mpg</MediaUri>
        </MediaLocator>
        <AnalyticEditingTemporalDecomposition gap="false" overlap="false">
          <Shot id="ClipS_3" locationReliability="0.9"
            editingLevelReliability="0.5" type="composite">
            <MediaTime>
              <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../MediaLocator[1]">35</MediaRelIncrTimePoint>
              <MediaIncrDuration
mediaTimeUnit="PT1N25F">413</MediaIncrDuration>
            </MediaTime>
          </Shot>
        </AnalyticEditingTemporalDecomposition>
      </AnalyticEditedVideo>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

In the second example, the shot is temporally decomposed into several composition shots; and the composition shots are temporally decomposed into intra-composition shots. The edition areas in the shot, composition shots and intra-composition shots are described using the EditedMovingRegion DS as the result of the spatio-temporal decomposition of the analytic edited video segments, i.e., using the AnalyticEditedVideoSegmentSpatioTemporalDecomposition DS. In composition and internal transition descriptions, the edited moving regions introduced or affected by the transitions are referenced by using the EditedMovingRegionRef element in the AnalyticTransition DS.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AnalyticEditedVideoType">
      <AnalyticEditedVideo xsi:type="EditedVideoType">
        <MediaLocator>
          <MediaUri>video.mpg</MediaUri>
        </MediaLocator>
        <AnalyticEditingTemporalDecomposition gap="false" overlap="false">
          <!-- Description of Eyexam -->
          <Shot id="ClipS_3" locationReliability="0.9"
            editingLevelReliability="0.5" type="composite">
            <MediaTime>
              <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../MediaLocator[1]">35</MediaRelIncrTimePoint>
              <MediaIncrDuration
mediaTimeUnit="PT1N25F">413</MediaIncrDuration>
            </MediaTime>
            <AnalyticEditionAreaDecomposition gap="false" overlap="true"
              criteria="edition area">
              <EditedMovingRegion id="editArea_1">
                <TextAnnotation>
                  <FreeTextAnnotation> Frame area </FreeTextAnnotation>
                </TextAnnotation>
                <SpatioTemporalLocator>
                  <FigureTrajectory type="rectangle">
                    <MediaTime>
                      <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../MediaLocator[1]">35</MediaRelIncrTimePoint>

```

```

        <MediaIncrDuration
          mediaTimeUnit="PT1N25F">413</MediaIncrDuration>
</MediaTime>
<Vertex>
  <WholeInterval>
    <MediaDuration>PT0S</MediaDuration>
  </WholeInterval>
  <InterpolationFunctions>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
  </InterpolationFunctions>
</Vertex>
<Vertex>
  <WholeInterval>
    <MediaDuration>PT1S</MediaDuration>
  </WholeInterval>
  <InterpolationFunctions>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
  </InterpolationFunctions>
</Vertex>
<Vertex>
  <WholeInterval>
    <MediaDuration>PT2S</MediaDuration>
  </WholeInterval>
  <InterpolationFunctions>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
  </InterpolationFunctions>
</Vertex>
  <!-- more elements here -->
</FigureTrajectory>
</SpatioTemporalLocator>
<RegionDepthType href="...">
  <Name xml:lang="en">background</Name>
</RegionDepthType>
</EditedMovingRegion>
<EditedMovingRegion id="editArea_2">
  <TextAnnotation>
    <FreeTextAnnotation>CenteredCercle</FreeTextAnnotation>
  </TextAnnotation>
  <SpatioTemporalLocator>
    <FigureTrajectory type="rectangle">
      <MediaTime>
        <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../../../../../MediaLocator[1]">175</MediaRelIncrTimePoint>
        <MediaIncrDuration
          mediaTimeUnit="PT1N25F">273</MediaIncrDuration>
      </MediaTime>

```



```

<Vertex>
  <WholeInterval>
    <MediaDuration>PT0S</MediaDuration>
  </WholeInterval>
  <InterpolationFunctions>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
  </InterpolationFunctions>
</Vertex>
<Vertex>
  <WholeInterval>
    <MediaDuration>PT1S</MediaDuration>
  </WholeInterval>
  <InterpolationFunctions>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
  </InterpolationFunctions>
</Vertex>
<Vertex>
  <WholeInterval>
    <MediaDuration>PT2S</MediaDuration>
  </WholeInterval>
  <InterpolationFunctions>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
    <KeyValue type="startPoint">
      <!-- more elements here -->
    </KeyValue>
  </InterpolationFunctions>
</Vertex>
  <!-- more elements here -->
</FigureTrajectory>
</SpatioTemporalLocator>
<RegionDepthType href="...">
  <Name xml:lang="en">foreground</Name>
</RegionDepthType>
</EditedMovingRegion>
</AnalyticEditionAreaDecomposition>
<AnalyticEditingTemporalDecomposition gap="false"
  overlap="false">
  <CompositionShot id="ClipCS_1" locationReliability="0.9"
    editingLevelReliability="0.5" type="simple">
    <MediaTime>
      <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">35</MediaRelIncrTimePoint>
      <MediaIncrDuration
        mediaTimeUnit="PT1N25F">55</MediaIncrDuration>
    </MediaTime>
    </CompositionShot>
    <CompositionTransition id="CT_1" locationReliability="0.9"
      editingLevelReliability="0.5" evolutionReliability="0.5">
      <MediaTime>

```

```

        <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">90</MediaRelIncrTimePoint>
        <MediaIncrDuration
            mediaTimeUnit="PT1N25F">10</MediaIncrDuration>
        </MediaTime>
        <EvolutionType href="...">
            <Name xml:lang="en">IrisOpen</Name>
        </EvolutionType>
        <!--Edition area introduced by composition transition -->
        <EditedMovingRegionRef href="#editArea_2"/>
    </CompositionTransition>
    <CompositionShot id="ClipCS_2" locationReliability="0.9"
        editingLevelReliability="0.5" type="composite">
        <MediaTime>
            <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">100</MediaRelIncrTimePoint>
            <MediaIncrDuration
                mediaTimeUnit="PT1N25F">348</MediaIncrDuration>
            </MediaTime>
            <AnalyticEditingTemporalDecomposition gap="false"
                overlap="true">
                <IntraCompositionShot id="ClipICS_1"
                    locationReliability="0.9"
                    editingLevelReliability="0.5" type="simple">
                    <MediaTime>
                        <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">100</MediaRelIncrTimePoint>
                        <MediaIncrDuration
                            mediaTimeUnit="PT1N25F">75</MediaIncrDuration>
                        </MediaTime>
                    </IntraCompositionShot>
                    <InternalTransition id="IT_1"
locationReliability="0.9"
                        editingLevelReliability="0.5"
                        evolutionReliability="0.5">
                        <MediaTime>
                            <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">175</MediaRelIncrTimePoint>
                            <MediaIncrDuration
                                mediaTimeUnit="PT1N25F">1</MediaIncrDuration>
                            </MediaTime>
                            <EvolutionType href="...">
                                <Name xml:lang="en">cut</Name>
                            </EvolutionType>
                            <!--Edition area affected by inter. transition -->
                            <EditedMovingRegionRef href="#editArea_2"/>
                        </InternalTransition>
                        <IntraCompositionShot id="ClipICS_2"
                            locationReliability="0.9"
                            editingLevelReliability="0.5" type="simple">
                            <MediaTime>
                                <MediaRelIncrTimePoint mediaTimeUnit="PT1N25F"
mediaTimeBase="../../../../../../../../MediaLocator[1]">175</MediaRelIncrTimePoint>
                                <MediaIncrDuration
                                    mediaTimeUnit="PT1N25F">273</MediaIncrDuration>
                                </MediaTime>
                            </IntraCompositionShot>
                        </AnalyticEditingTemporalDecomposition>
                    </CompositionShot>
                </AnalyticEditingTemporalDecomposition>
    </CompositionShot>
</AnalyticEditingTemporalDecomposition>

```

```

        </Shot>
    </AnalyticEditingTemporalDecomposition>
</AnalyticEditedVideo>
</MultimediaContent>
</Description>
</Mpeg7>

```

11.10 Structural relation classification schemes

11.10.1 Introduction

This subclause specifies classification schemes for describing temporal and spatial relations. Table 26 provides an overview of the temporal and spatial relation classification schemes. In the following specification of the TemporalRelation CS and SpatialRelation CS, symbol *B* refers to the source of the relation and symbol *C* refers to the target of the relation.

Table 26 — Overview of the classification schemes for structural relations

Tools	Functionality
TemporalRelation CS	This classification scheme specifies temporal relations. The temporal relations apply to entities that have time attributes, which may be implicit or explicitly stated.
SpatialRelation CS	This classification scheme specifies spatial relations. The spatial relations apply to entities that have 2-D spatial attributes, which may be implicit or explicitly stated.

11.10.2 TemporalRelation CS

11.10.2.1 Introduction

The TemporalRelation CS defines temporal relations. The temporal relations apply to entities that have time attributes, which may be implicit or explicitly stated. For example, the temporal relations can be used to describe the temporal arrangement of video segments or the order of semantic events in time.

11.10.2.2 TemporalRelation CS syntax

```

<!-- ##### -->
<!-- Definition of TemporalRelation CS (11.10.2) -->
<!-- ##### -->

<ClassificationScheme uri="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001">
  <Header xsi:type="DescriptionMetadataType">
    <Comment>
      <FreeTextAnnotation xml:lang="en">
        Temporal relations
        In defining these relations, each argument is represented by the
        smallest temporal interval that includes the extent of the argument in the time
        dimension. For each B, the start and end times are denoted by B.a (start) and B.b
        (end), respectively.
      </FreeTextAnnotation>
    </Comment>
  </Header>

  <Term termID="precedes">
    <Definition>
      B precedes C if and only if B ends before C starts.
    </Definition>
  </Term>
</ClassificationScheme>

```

```

    </Definition>
  </Term>
  <Term termID="follows">
    <Definition> Inverse relation of precedes </Definition>
  </Term>
  <Term termID="meets">
    <Definition>
      B meets C if and only if B ends at the same time as C starts.
    </Definition>
  </Term>
  <Term termID="metBy">
    <Definition> Inverse relation of meets </Definition>
  </Term>
  <Term termID="overlaps">
    <Definition>
      B overlaps C if and only if B starts before C starts, B ends after C
      starts and B ends before C ends.
    </Definition>
  </Term>
  <Term termID="overlappedBy">
    <Definition> Inverse relation of overlaps </Definition>
  </Term>
  <Term termID="contains">
    <Definition>
      B contains C if and only if C starts after B starts and C ends at the same
      time as or before B ends, or C starts after or at the same time as B starts and C
      ends before B ends.
    </Definition>
  </Term>
  <Term termID="during">
    <Definition> Inverse relation of contains </Definition>
  </Term>
  <Term termID="strictContains">
    <Definition>
      B strictContains C if and only if C starts after B starts and C ends
      before B ends.
    </Definition>
  </Term>
  <Term termID="strictDuring">
    <Definition> Inverse relation of strictContains </Definition>
  </Term>
  <Term termID="starts">
    <Definition>
      B starts C if and only if B starts at the same time as C starts and B ends
      before C ends.
    </Definition>
  </Term>
  <Term termID="startedBy">
    <Definition> Inverse relation of starts </Definition>
  </Term>
  <Term termID="finishes">
    <Definition>
      B finishes C if and only if B starts after C starts and B ends at the same
      time as C ends.
    </Definition>
  </Term>
  <Term termID="finishedBy">
    <Definition> Inverse relation of finishes </Definition>
  </Term>
  <Term termID="coOccurs">

```

```

    <Definition>
        B coOccurs if and only if B starts at the same time as C starts and B ends
        at the same time as C ends.
    </Definition>
</Term>

<Term termID="contiguous">
    <Definition>
        A1, A2, ... An contiguous if and only if they are temporally disjoint and
        connected.
    </Definition>
</Term>
<Term termID="sequential">
    <Definition>
        A1, A2, ... An sequential if and only if they are temporally disjoint and
        not necessarily connected.
    </Definition>
</Term>
<Term termID="coBegin">
    <Definition>
        A1, A2, ... An coBegin if and only if they start at the same time.
    </Definition>
</Term>
<Term termID="coEnd">
    <Definition>
        A1, A2, ... An coEnd if and only if they end at the same time.
    </Definition>
</Term>
<Term termID="parallel">
    <Definition>
        A1, A2, ... An parallel if and only if the intersection of A1, A2, ... An has
        one non-empty interior.
    </Definition>
</Term>
<Term termID="overlapping">
    <Definition>
        A1, A2, ... An overlapping if and only if the union of A1, A2, ... An is
        connected and each Ai intersects at least one other Aj with non-empty interior.
    </Definition>
</Term>
</ClassificationScheme>

```

11.10.2.3 TemporalRelation CS semantics

Table 27 and Table 28 define the semantics of the temporal relations. Each entry in Table 27 identifies the name of the relation and its inverse relation, defines the relation mathematically, describes its properties (optional), and gives examples of its use. Each entry in Table 28 identifies the name of the relation, defines the relation mathematically, and gives examples of its use.

In defining the temporal relations, each argument is represented by the smallest temporal interval that includes the extent of the argument in the time dimension. For each B , the start and end times are denoted by $B.a$ (start) and $B.b$ (end), respectively. The inverse relations are defined as follows: " B Relation C " is equivalent to " C InverseRelation B ", where B is the source and C is the target of the *Relation*, and C is the source and B is the target of the *InverseRelation*. This definition applies also in the case of each n -ary relation that has an inverse relation, where $B=(b1, ..., bn)$ is the n -tuple of sources and $C=(c1, ..., cm)$ is the m -tuple of the targets of the *Relation*.

Table 27 — Semantics of relations in the TemporalRelation CS (binary relations)

Relation Name	Inverse Relation	Definition	Properties	Examples (informative)
precedes	follows	B precedes C if and only if $B.b < C.a$	Transitive	BBB CCC
meets	metBy	B meets C if and only if $B.b = C.a$	Anti-symmetric	BBBCCC
overlaps	overlappedBy	B overlaps C if and only if $B.a < C.a$ AND $B.b > C.a$ AND $B.b < C.b$		BBB CCC
contains	during	B contains C if and only if $(C.a > B.a$ AND $C.b \leq B.b)$ OR $(C.a \geq B.a$ AND $C.b < B.b)$	Transitive	Any of the examples for strictContains, startedBy, and finishedBy.
strictContains	strictDuring	B strictContains C if and only if $C.a > B.a$ AND $C.b < B.b$	Transitive	BBBBBBB CCCC
starts	startedBy	B starts C if and only if $B.a = C.a$ AND $B.b < C.b$	Transitive	BBBB CCCCC
finishes	finishedBy	B finishes C if and only if $B.a > C.a$ AND $B.b = C.b$	Transitive	BBBB CCCCC
coOccurs	coOccurs	B coOccurs C if and only if $B.a = C.a$ AND $B.b = C.b$	Equivalence	BBB CCC

Table 28 — Semantics of relations in the TemporalRelation CS (n-ary relations)

Relation Name	Definition	Examples (informative)
contiguous	$A_1, A_2, \dots A_n$ contiguous if and only if $A_i.b = A_{i+1}.a$ for $i=1, \dots, n-1$ That is, $A_1, A_2, \dots A_n$ contiguous if and only if they are temporally disjoint and connected.	$A_1A_1A_1A_2A_2\dots A_nA_nA_n$
sequential	$A_1, A_2, \dots A_n$ sequential if and only if $A_i.b \leq A_{i+1}.a$ for $i=1, \dots, n-1$ That is, $A_1, A_2, \dots A_n$ sequential if and only if they are temporally disjoint and not necessarily connected.	$A_1A_1A_1$ $A_2A_2\dots A_nA_nA_n$
coBegin	$A_1, A_2, \dots A_n$ coBegin If and only if $A_i.a = A_{i+1}.a$ for $i=1, \dots, n-1$ That is, $A_1, A_2, \dots A_n$ coBegin if and only if they start at the same time.	$A_1A_1A_1$ A_2A_2 ... $A_nA_nA_n$
coEnd	$A_1, A_2, \dots A_n$ coEnd if and only if $A_i.b = A_{i+1}.b$ for $i=1, \dots, n-1$ That is, $A_1, A_2, \dots A_n$ coEnd if and only if they end at the same time.	$A_1A_1A_1$ A_2A_2 ... $A_nA_nA_n$
parallel	$A_1, A_2, \dots A_n$ parallel if and only if the intersection of $A_1, A_2, \dots A_n$ has one non-empty interior.	$A_1A_1A_1$ A_2A_2 ... $A_nA_nA_n$
overlapping	$A_1, A_2, \dots A_n$ overlapping if and only if the union of $A_1, A_2, \dots A_n$ is connected and each A_i intersects at least one other A_i with non-empty interior.	$A_1 A_1 A_1$ $A_2A_2A_2A_2$... $A_nA_nA_n$

11.10.2.4 TemporalRelation CS examples (informative)

The following example illustrates the use of the TemporalRelation CS and the Graph DS for describing the relation that video segment A precedes video segment B.

```
<VideoSegment id="segmentA"> <!-- more elements here --> </VideoSegment>
<VideoSegment id="segmentB"> <!-- more elements here --> </VideoSegment>
<Graph>
  <Node id="nodeA" href="#segmentA"/>
  <Node id="nodeB" href="#segmentB"/>
  <!-- Edge a->b -->
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes"
    source="#nodeA" target="#nodeB"/>
</Graph>
```

The following example illustrates the use of the TemporalRelation CS and the Relation DS for describing the relation that video segment C precedes video segment D.

```
<VideoSegment id="segmentC">
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes"
    target="#segmentD"/>
  <MediaTime> <!-- more elements here --> </MediaTime>
</VideoSegment>

<VideoSegment id="segmentD">
  <MediaTime> <!-- more elements here --> </MediaTime>
</VideoSegment>
```

The following example illustrates the use of the TemporalRelation CS and the Graph DS for describing temporal relations between the narrator moving region ("narrator-movreg"), the rabbit moving region ("rabbit-movreg"), and the sun moving region ("sun-movreg") of a documentary video on the subject of nature. It is assumed that the segments have been described using the Segment DSs with the specified id attributes. Two variations are shown as follows, the first uses Relation and Node in the Graph DS, and the second uses only Relation:

```
<MovingRegion id="sun-movreg">
  <SpatioTemporalLocator> <!-- more elements here --> </SpatioTemporalLocator>
</MovingRegion>
<MovingRegion id="narrator-movreg">
  <SpatioTemporalLocator> <!-- more elements here --> </SpatioTemporalLocator>
</MovingRegion>
<MovingRegion id="rabbit-movreg">
  <SpatioTemporalLocator> <!-- more elements here --> </SpatioTemporalLocator>
</MovingRegion>

<!-- First variation of the temporal graph -->
<Graph type="temporal">
  <Node id="node1" href="#sun-movreg"/>
  <Node id="node2" href="#rabbit-movreg"/>
  <Node id="node3" href="#narrator-movreg"/>
  <Node id="node4" href="#sun-movreg"/>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:starts"
    source="#node2" target="#node1"/>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes"
    source="#node3" target="#node4"/>
</Graph>

<!-- Second variation of the temporal graph -->
```



```

<Graph type="temporal">
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:starts"
    source="#rabbit-movreg" target="#sun-movreg"/>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes"
    source="#narrator-movreg" target="#sun-movreg"/>
</Graph>

```

11.10.3 SpatialRelation CS

11.10.3.1 Introduction

The SpatialRelation CS defines spatial relations. The spatial relations apply to entities that have 2-D spatial attributes, which may be implicit or explicitly stated. For example, the spatial relations can be used to describe the spatial arrangement of regions in an image or the spatial arrangement of semantic objects.

11.10.3.2 SpatialRelation CS syntax

```

<!-- ##### -->
<!-- Definition of SpatialRelation CS (11.10.3) -->
<!-- ##### -->

<ClassificationScheme uri="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001">
  <Header xsi:type="DescriptionMetadataType">
    <Comment>
      <FreeTextAnnotation xml:lang="en">
        Spatial relations
        In defining the spatial relations, each argument (B or C) is
        represented by the smallest rectangular 2-D bounding box that includes the spatial
        extent of the argument in the 2-D spatial dimensions. For each B, the x and y
        spatial coordinates are represented by B.x and B.y, respectively. Furthermore, for
        spatial dimension x, the start and end points are denoted by B.x.a (start) and B.x.b
        (end), respectively and for similarly for spatial dimension y.
      </FreeTextAnnotation>
    </Comment>
  </Header>

  <Term termID="south">
    <Definition>
      B south C if and only if, in the dimension x, B starts after or at the
      same time as C starts and B ends at the same time as or before C ends, or B starts at
      the same time as or before C starts and B ends after or at the same time as C ends;
      and, in the dimension y, B ends at the same time as or before C starts.
    </Definition>
  </Term>
  <Term termID="north">
    <Definition> Inverse relation of south </Definition>
  </Term>
  <Term termID="west">
    <Definition>
      B west C if and only if, in the dimension x, B ends at the same time as or
      before C starts; and, in the dimension y, B starts after or at the same time as C
      starts and B ends at the same time as or before C ends, or B starts at the same time
      as or before C starts and B ends after or at the same time as C ends.
    </Definition>
  </Term>
  <Term termID="east">
    <Definition> Inverse relation of west </Definition>
  </Term>

```

```

<Term termID="northwest">
  <Definition>
    B northwest C if and only if, in the dimension x, B ends at the same time
    as or before C starts; and, in the dimension y, B starts after or at the same time as
    C ends.
  </Definition>
</Term>
<Term termID="southeast">
  <Definition> Inverse relation of northwest </Definition>
</Term>
<Term termID="southwest">
  <Definition>
    B southwest C if and only if, in the dimension x, B ends at the same time
    as or before C starts; and, in the dimension y, B ends at the same time as or before
    C starts.
  </Definition>
</Term>
<Term termID="northeast">
  <Definition> Inverse relation of southwest </Definition>
</Term>
<Term termID="left">
  <Definition>
    B left C if and only if, in the dimension x, B ends at the same time as or
    before C starts.
  </Definition>
</Term>
<Term termID="right">
  <Definition> Inverse relation of left </Definition>
</Term>
<Term termID="below">
  <Definition>
    B below C if and only if, in the dimension y, B ends at the same time as
    or before C starts.
  </Definition>
</Term>
<Term termID="above">
  <Definition> Inverse relation of below </Definition>
</Term>
<Term termID="over">
  <Definition>
    B over C if and only if, in the dimension x, B starts at the same time as
    or before C starts and B ends after C starts, or B starts after C starts and B starts
    before C ends; and, in the dimension y, B starts at the same time as C ends.
  </Definition>
</Term>
<Term termID="under">
  <Definition> Inverse relation of over </Definition>
</Term>
</ClassificationScheme>

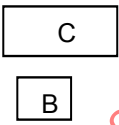
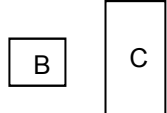
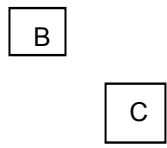
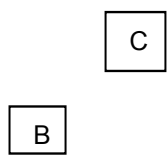
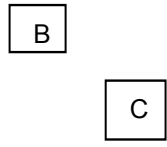
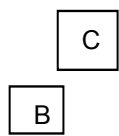
```

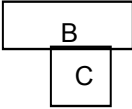
11.10.3.3 SpatialRelation CS semantics

Table 29 define the semantics of the relations in the SpatialRelation CS. Each entry in Table 29 identifies the name of the relation and its inverse relation, defines the relation mathematically, describes its properties (optional), and gives examples of its use. In defining the spatial relations, each argument (*B* or *C*) is represented by the smallest rectangular 2-D bounding box that includes the spatial extent of the argument in the 2-D spatial dimensions. For each *B*, the *x* and *y* spatial coordinates are represented by *B.x* and *B.y*, respectively. Furthermore, for spatial dimension *x*, the start and end points are denoted by *B.x.a* (start) and *B.x.b* (end), respectively and for similarly for spatial dimension *y*. The inverse relations are defined as follows: "*B Relation C*" is equivalent to "*C*

InverseRelation B', where *B* is the source and *C* is the target of the *Relation*, and *C* is the source and *B* is the target of the *InverseRelation*.

Table 29 — Semantics of relations in the SpatialRelation CS

Relation Name	Inverse Relation	Definition	Properties	Informative Examples
south	north	B south C if and only if ((B.x.a ≥ C.x.a AND B.x.b ≤ C.x.b) OR (B.x.a ≤ C.x.a AND B.x.b ≥ C.x.b)) AND B.y.b ≤ C.y.a	Transitive	
west	east	B west C if and only if B.x.b ≤ C.x.a AND ((B.y.a ≥ C.y.a AND B.y.b ≤ C.y.b) OR (B.y.a ≤ C.y.a AND B.y.b ≥ C.y.b))	Transitive	
northwest	southeast	B northwest C if and only if B.x.b ≤ C.x.a AND B.y.a ≥ C.y.b	Transitive	
southwest	northeast	B southwest C if and only if B.x.b ≤ C.x.a AND B.y.b ≥ C.y.a	Transitive	
left	right	B left C if and only if B.x.b ≤ C.x.a	Transitive	
below	above	B below C if and only if B.y.b ≤ C.y.a	Transitive	

Relation Name	Inverse Relation	Definition	Properties	Informative Examples
over	under	B over C if and only if $((B.x.a \leq C.x.a \text{ AND } B.x.b > C.x.a) \text{ OR } (B.x.a > C.x.a \text{ AND } B.x.a < C.x.b))$ AND $B.y.a = C.y.b$	Transitive	

11.10.3.4 SpatialRelation CS examples (informative)

The following example illustrates the use of the SpatialRelation CS for describing spatial relations among segments. The 2D-String is a data structure for spatial reasoning. In the following example, the 2D-string describes the spatial relations between the regions. In Figure 51, the regions correspond to the rabbit ("rabbit-rg"), cloud ("cloud-rg"), airplane ("airplane-rg"), and car ("car-rg"). The 2D-String is given as follows: (rabbit < cloud < car = airplane, rabbit = car < airplane < cloud), where the symbol "=" denotes the spatial relations "at the same x as" and "at the same y as", the symbol "<" denotes the spatial relations "left/right" and "below/above", and the symbol ":" denotes the spatial relation "in the same set as". The description of the associated 2D-String for the example in Figure 51 is included below. All the relations in this example are standard relations except for "sameX" and "sameY".

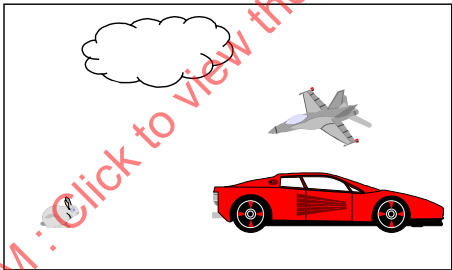


Figure 51 — Example description of spatial relations in an image using 2D string

```
<!-- 2D String rabbit<cloud<car=airplane, rabbit=car<airplane<cloud -->
<Graph>
  <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:left"
    source="#rabbit-rg" target="#cloud-rg"/>
  <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:left"
    source="#cloud-rg" target="#car-rg"/>
  <Relation name="urn:example:arelacioncs:sameX"
    source="#airplane-rg" target="#car-rg"/>
  <Relation name="urn:example:arelacioncs:sameY"
    source="#car-rg" target="#rabbit-rg"/>
  <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:below"
    source="#car-rg" target="#airplane-rg"/>
  <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:below"
    source="#airplane-rg" target="#cloud-rg"/>
</Graph>
```

11.10.4 Structural relation classification schemes examples (informative)

The following example illustrates the use of the TemporalRelation CS and the SpatialRelation CS for describing structural relations among segments of a video segment capturing a goal in a soccer game. The video segment corresponds to *id* = "Goal-sg". The kick and not-catch events in the video shot are represented by video segments *id* = "Kick-sg" and *id* = "Not-Catch-sg", respectively. The spatio-temporal regions corresponding to the goal, the ball, the forward, and the goalkeeper are described by moving regions *id* = "Goal-rg", *id* = "Ball-rg", "Forward-rg", and *id* = "Goalkeeper-rg", respectively.

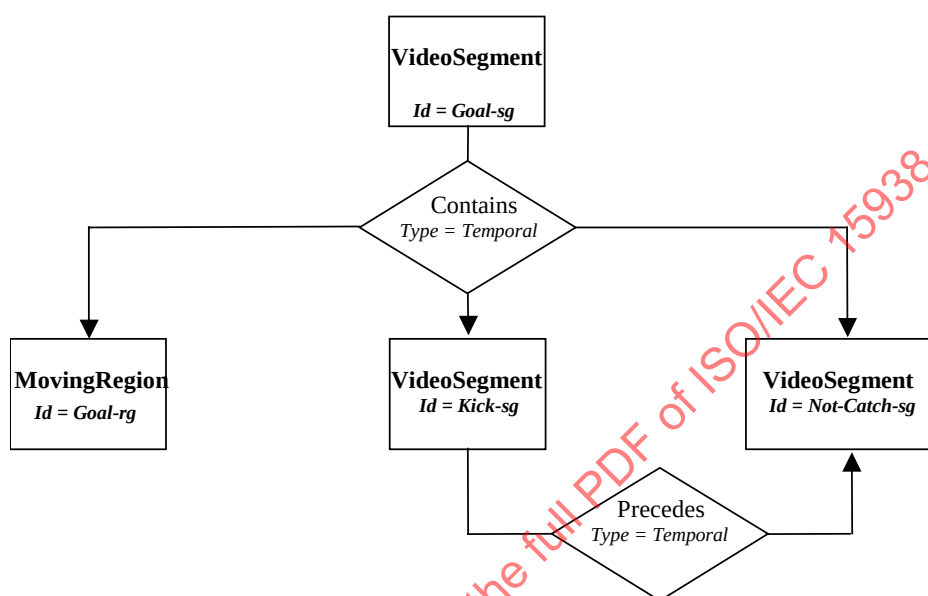


Figure 52 — Examples of relationships among video segments and moving regions

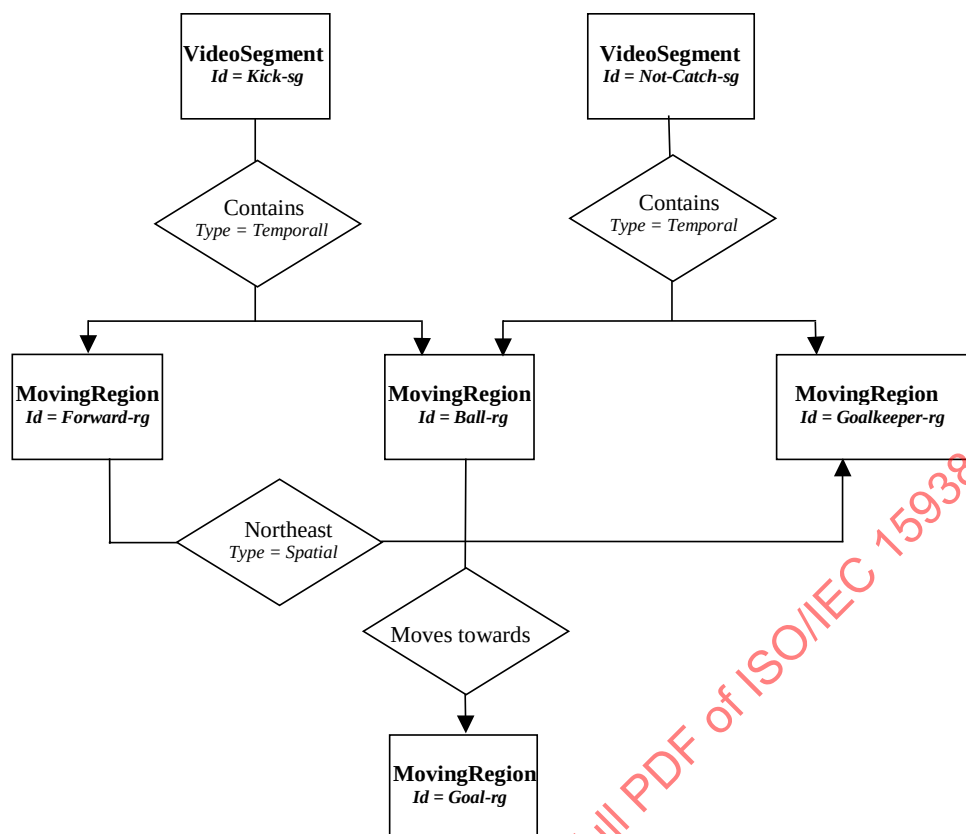


Figure 53 — Examples of relationships among video segments and moving regions

The Graph DS describes the relations between the video segments and moving regions. Examples of relations between the segments are shown in Figure 52 and Figure 53. In Figure 52 (Graph 1), the region id = "Goal-rg", the video segment id = "Kick-sg", and the video segment id = "Not-Catch-sg" are described as being temporally contained in the segment id = "Goal-sg". The video segment id = "Kick-sg" is also described as preceding the video segment id = "Not-Catch-sg".

In Figure 53 (Graph 2), the video segment id = "Kick-sg" is described as temporally containing the moving regions id = "Forward-rg" and the id = "Ball-rg". Similarly, the video segment id = "Not-Catch-sg" is described as temporally containing the moving regions id = "Ball-rg" and the id = "Goalkeeper-rg". The region id = "Forward-rg" is related to the moving region id = "Goalkeeper-rg" by a spatial relation, "Northeast". The moving region id = "Ball-rg" is related to the moving region id = "Goal-rg" by a visual, global relation, "Moves towards".

The description of the segment graphs shown in Figure 52 and Figure 53 is included below.

```

<!-- Description of Graph 1 -->
<Graph>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains"
    source="#Goal-sg" target="#Goal-rg"/>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains"
    source="#Goal-sg" target="#Kick-sg"/>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains"
    source="#Goal-sg" target="#Not-Catch-sg"/>
  <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes"
    source="#Kick-sg" target="#Not-Catch-sg"/>
</Graph>

<!-- Description of Graph 2 -->
<Graph>

```

```

<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains"
  source="#Kick-sg" target="#Forward-rg"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains"
  source="#Kick-sg" target="#Ball-rg"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains"
  source="#Not-Catch-sg" target="#Ball-rg"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains"
  source="#Not-Catch-sg" target="#Goalkeeper-rg"/>
<Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast"
  source="#Forward-rg" target="#Goalkeeper-rg"/>
<Relation name="urn:example:arelacioncs:movesToward"
  source="#Ball-rg" target="#Goal-rg"/>
</Graph>

```

12 Semantics description tools

12.1 Introduction

This clause specifies tools for describing the semantics of multimedia content. These tools can be used to describe real-life concepts or narratives, which includes objects, agent objects, events, concepts, states, places, times, and narrative worlds, which are depicted by or related to multimedia content. These tools can also be used to describe semantic attributes and semantic relations.

12.1.1 Overview

The tools specified in this clause include the following:

Table 30 — Overview of the semantics description tools

<i>Tool</i>	<i>Functionality</i>
Semantic entity description tools	Describes semantic entities such as objects, agent objects, events, concepts, states, places, times, and narrative worlds.
Semantic attribute description tools	Describes attributes of semantic entities related to abstraction in semantic descriptions and semantic measurements in time and space.
Semantic relation classification schemes	Describes semantic relations.

Figure 54 illustrates the organization of the tools for describing the semantics of multimedia content.

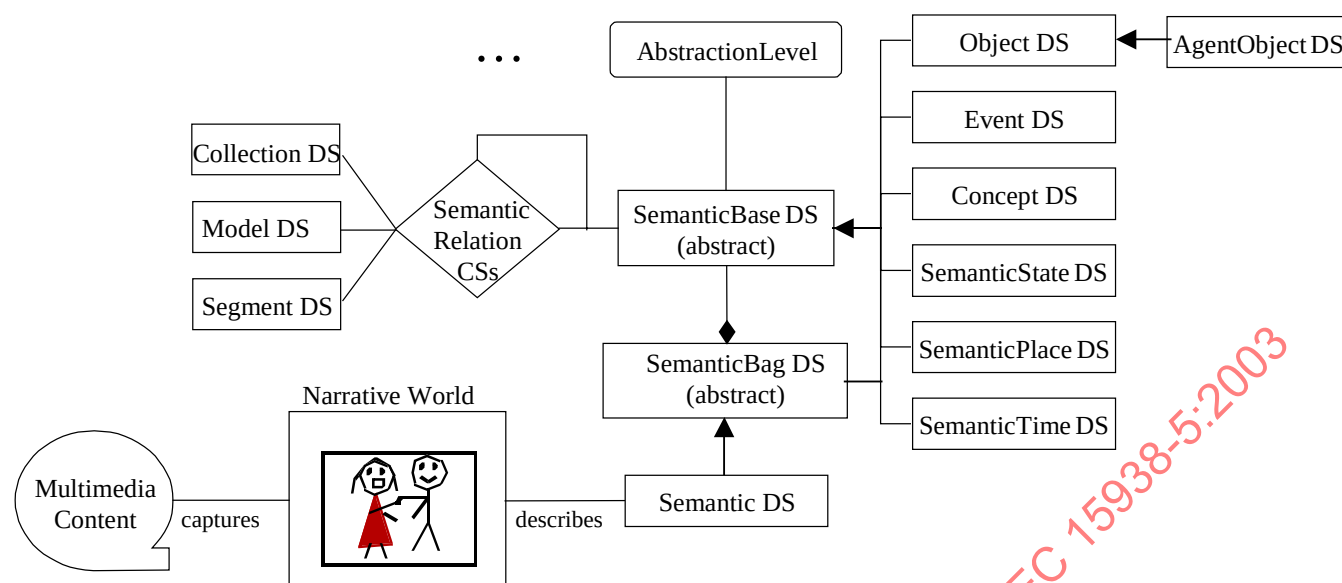


Figure 54 — Illustration of the relationships of the tools for describing the semantics of multimedia content

The principal concepts are defined as follows:

- **Narrative world:** refers to information that makes up a narrative or real-life. The information includes the background, context, and participants that are depicted by or related to multimedia content. An instance of multimedia content may have multiple semantics, and vice versa. Entities can change status or functionality in narrative worlds, and, therefore, be described by different semantic descriptions. For instance, a picture can be an object (when it hangs in a museum) or a location ("In the picture"), or can become the narrative world itself (when describing the scene the picture depicts "In the picture Caesar is talking to Brutus").
- **Abstraction:** refers to the process of generalizing a semantic description of an instance of multimedia content (a concrete description) to a semantic description of a set of instances of multimedia content (a media abstraction), or to a semantic description of a set of concrete semantic descriptions (a formal abstraction). The abstraction model for the semantic entity description tools is described in 12.2.

As shown in Figure 54, the SemanticBase DS is an abstract tool that is the base of the tools that describe semantic entities. The specialized semantic entity description tools extend the base SemanticBase DS to describe specific types of semantic entities, such as objects, agent objects, events, concepts, states, places, times, and narrative worlds.

The semantic entity description tools are summarized as follows:

Tool	Functionality
SemanticBase DS	Abstract base tool for describing semantic entities.
SemanticBag DS	Abstract base tool for describing collections of semantic entities and their relations.
Semantic DS	Describes narrative worlds that are depicted by or related to multimedia content.
Object DS	Describes objects in narrative worlds.
AgentObject DS	Describes objects that are persons, organizations, or groups of people in narrative worlds.

Event DS	Describes events in narrative worlds.
Concept DS	Describes abstract semantic entities that cannot be described as abstractions of any objects, events, times, places, or states.
SemanticState DS	Describes states or parametric attributes of semantic entities and semantic relations, at a given time or location in narrative worlds, or in given media locations.
SemanticPlace DS	Describes locations in narrative worlds.
SemanticTime DS	Describes times in narrative worlds.

The semantic attribute description tools describe attributes of semantic entities related to abstraction in semantic descriptions and semantic measurements in time or space.

The semantic attribute description tools are summarized as follows:

<i>Tool</i>	<i>Functionality</i>
AbstractionLevel datatype	Describes the abstraction performed in the description of a semantic entity.
Extent datatype	Describes the extent or size semantic attribute.
Position datatype	Describes the position semantic attribute.

The semantic relation classification schemes defines semantic relations. The semantic relation classification schemes are summarized as follows:

<i>Classification Scheme</i>	<i>Functionality</i>
SemanticRelation CS	Describes semantic relations such as the relationship of events in a narrative, or relationship of an object to multimedia content that depicts the object.

12.1.2 Semantics description example

Figure 55 shows an informative example of how the semantic description tools describe the semantics of an image. The boxes in Figure 55 represent descriptions of different semantic entity description tools and semantic relation classification schemes.

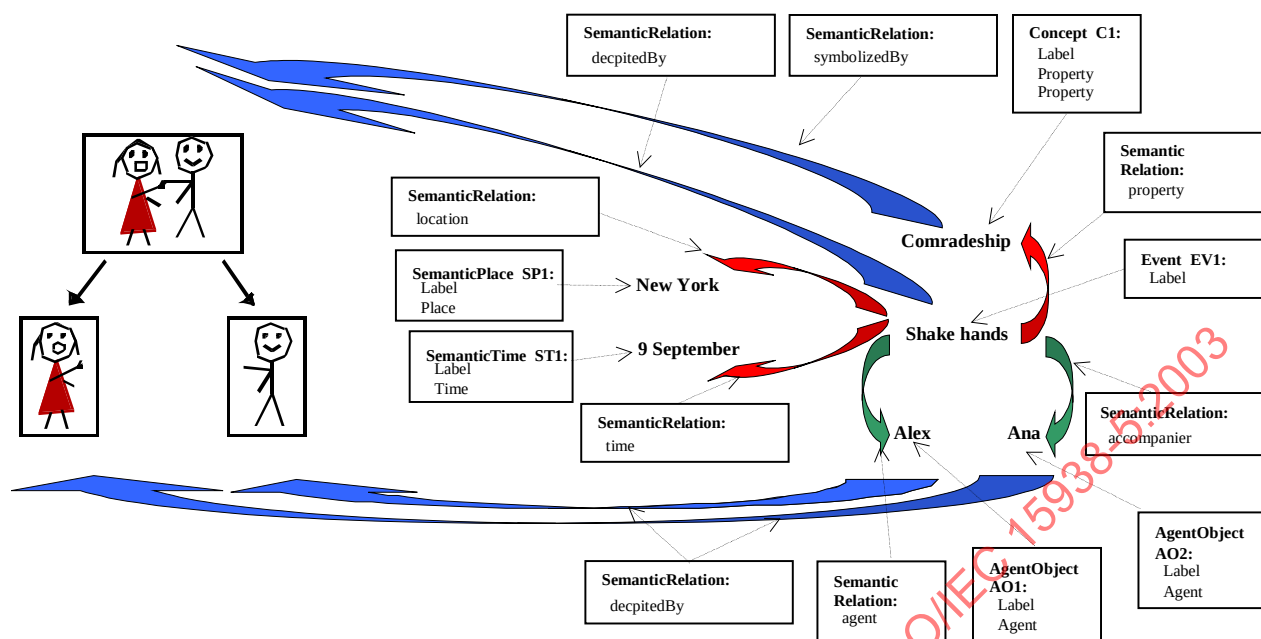


Figure 55 — Illustration of the description of a narrative world captured by an image using tools for describing the semantics of multimedia content

The semantic description of the image is the following: "Alex is shaking hands with Ana in New York on the 9th of September. The event is showing "comradeship". In this example, the AgentObject DS, the Event DS, the SemanticPlace DS, the SemanticTime DS, and the Concept DS describe people, event, place, time and concept of the image, respectively. A Label element describes each semantic entity; the Property, Time, Place and Agent elements are also used to describe the attributes of people, event, place, time and concept of the image. The SemanticRelation CS describes the relations among semantic entities and possibly other non-semantic entities.

12.2 Abstraction model

The semantics description tools describe the semantics of instances or abstractions of multimedia content. The kinds of abstractions in descriptions are described using the AbstractionLevel datatype (see 12.4.2). Other abstract quantities such as properties and concepts are described using the Property element in SemanticBase DS (see 12.3.2) and the Concept DS (see 12.3.8), respectively. The abstraction model is defined as follows:

- A concrete description describes the semantics of an instance of multimedia content such as an image or a video segment. Figure 55 shows the concrete description of an image.
- An abstraction, or a lambda abstraction, in logic, creates a function from a statement by replacing one or more of the constant expressions in the statement by a variable. Abstraction thus replaces a single instance with a more general class.
- A media abstraction is a semantic description that has been separated from a specific instance of multimedia content, and describes all instances of multimedia content that are similar. A media abstraction of the description in Figure 55 is the description "Alex is shaking hands with Ana in New York on the 9th of September", with no links to the image (shown in Figure 56.a); this description is, therefore, applicable to any image or video depicting that event. The media is the variable of a media abstraction.
- A formal abstraction is a semantic description that contains semantic entities acting as variables common to a set of semantic descriptions. These variables, when filled in, create concrete semantic descriptions or media abstractions. A formal abstraction can be thought as a first order lambda abstraction. Some

examples of formal abstractions of the description in Figure 55 are the description "Alex is shaking hands with any woman in New York on the 9th of September", shown in Figure 56(b), and "Any man is shaking hands with any woman in New York on the 9th of September", where "any woman" and "any man" are the variables of the formal abstractions. The variables in a formal abstraction can be replaced or filled in with concrete instances using the semantic relation exemplifies.

- Properties are qualities of semantic entities (e.g., "ripeness" of a "banana"). Concepts are collections of properties that define a category of entities but cannot fully characterize it, i.e., abstract entities or qualities that are not the result of an abstraction such as "comradeship" in Figure 55. An abstract entity or quality is generally describable only through its properties, and therefore is not abstract because it comes from the replacement of elements of a description by variables.

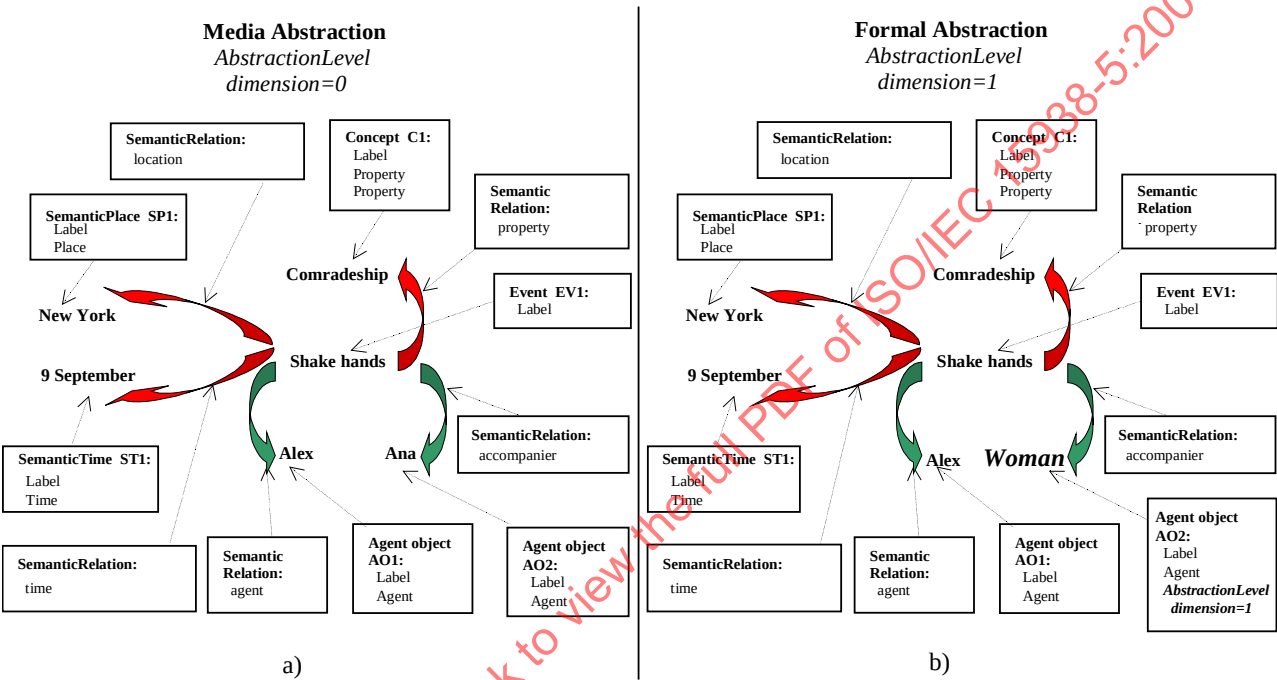


Figure 56 — Illustration of a media abstraction (a) and a formal abstraction (b) of the concrete description illustrated in Figure 55

12.3 Semantic entity description tools

12.3.1 Introduction

This subclause specifies tools for describing semantic entities depicted by, or related to multimedia content. The following table summarizes the tools specified in this subclause. Figure 54 shows the organization of the tools for describing semantic entities. Figure 55 shows an example of how the semantic entity description tools can be used to describe a narrative world.

Table 31 — Overview of the semantic entity description tools

Tool	Functionality
SemanticBase DS	This tool is an abstract tool that provides the base of the tools that describe a semantic entity depicted by, or related to multimedia content. The specialized semantic entity description tools extend the the SemanticBase DS.

<i>Tool</i>	<i>Functionality</i>
SemanticBag DS	This tool is an abstract tool that provides the base of the tools that describe a collection of semantic entities and their relations. This tool extends the SemanticBase DS.
Semantic DS	This tool describes a narrative world that is depicted by or related to the multimedia content. This tool extends the SemanticBag DS.
Object DS	This tool describes an object in a narrative world. This tool extends the SemanticBase DS. This tool extends the SemanticBase DS.
AgentObject DS	This tool describes an object that is a person, an organization, or a group of people in a narrative world. This tool extends the Object DS.
Event DS	This tool describes an event in a narrative world. This tool extends from the SemanticBase DS.
Concept DS	This tool describes an abstract semantic entity that cannot be described as an abstraction of any object, event, time, place, or state. This tool extends the SemanticBase DS.
SemanticState DS	This tool describes a state or parametric attributes of a semantic entity or semantic relation, at a given time or location in the narrative world, or in a given media location. This tool extends the SemanticBase DS.
SemanticPlace DS	This tool describes location in a narrative world. This tool extends the SemanticBase DS.
SemanticTime DS	This tool describes time in a narrative world. This tool extends the SemanticBase DS.

12.3.2 SemanticBase DS

12.3.2.1 Introduction

The SemanticBase DS is an abstract type from which the specialized semantic entity description tools are derived. The SemanticBase DS defines the properties of semantic entities, such as labels and relations, that apply to the specialized semantic entity description tools: SemanticBag DS, Semantic DS, Object DS, AgentObject DS, Event DS, Concept DS, SemanticState DS, SemanticPlace DS, and SemanticTime DS.

12.3.2.2 SemanticBase DS syntax

```

<!-- ##### -->
<!-- Definition of SemanticBase DS (12.3.2) -->
<!-- ##### -->

<!-- Definition of SemanticBase DS -->
<complexType name="SemanticBaseType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="AbstractionLevel" type="mpeg7:AbstractionLevelType"
          minOccurs="0"/>
        <element name="Label" type="mpeg7:TermUseType"
          minOccurs="1" maxOccurs="unbounded"/>

```

```

<element name="Definition" type="mpeg7:TextAnnotationType"
  minOccurs="0"/>
<element name="Property" type="mpeg7:TermUseType"
  minOccurs="0" maxOccurs="unbounded"/>
<element name="MediaOccurrence" minOccurs="0" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <choice minOccurs="0">
        <element name="MediaInformation"
          type="mpeg7:MediaInformationType"/>
        <element name="MediaInformationRef"
          type="mpeg7:ReferenceType"/>
        <element name="MediaLocator"
          type="mpeg7:MediaLocatorType"/>
      </choice>
      <element name="Mask" type="mpeg7:MaskType"
        minOccurs="0"/>
      <element name="AudioDescriptor" type="mpeg7:AudioDType"
        minOccurs="0" maxOccurs="unbounded"/>
      <element name="AudioDescriptionScheme"
        type="mpeg7:AudioDSType"
        minOccurs="0" maxOccurs="unbounded"/>
      <element name="VisualDescriptor" type="mpeg7:VisualDType"
        minOccurs="0" maxOccurs="unbounded"/>
      <element name="VisualDescriptionScheme"
        type="mpeg7:VisualDSType"
        minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="type" use="optional" default="perceivable">
      <simpleType>
        <union>
          <simpleType>
            <restriction base="NMTOKEN">
              <enumeration value="perceivable"/>
              <enumeration value="reference"/>
              <enumeration value="symbol"/>
            </restriction>
          </simpleType>
          <simpleType>
            <restriction base="mpeg7:termReferenceType"/>
          </simpleType>
        </union>
      </simpleType>
    </attribute>
  </complexType>
</element>
<element name="Relation" type="mpeg7:RelationType"
  minOccurs="0" maxOccurs="unbounded"/>
</sequence>
</extension>
</complexContent>
</complexType>

```

12.3.2.3 SemanticBase DS semantics

Semantics of the SemanticBaseType:

Name	Definition
SemanticBaseType	This type is an abstract type from which the specialized semantic entity description tools are derived. The SemanticBase DS describes a semantic entity (abstract). Examples of semantic entities are objects, events, and narrative worlds, among others. SemanticBaseType extends DType
AbstractionLevel	Indicates the kind of abstraction performed in the description of the semantic entity (optional). When it is not present, then the description is a concrete semantic description of actual multimedia content. If it is present, the description is a media abstraction or a formal abstraction (see 12.4.2).
Label	Describes the semantic entity. Any terms can be supplied for Label that identify or describe the semantic entity.
Definition	Defines the semantic entity (optional).
Property	Describes a quality or adjectival property associated with the semantic entity (optional).
MediaOccurrence	Describes an appearance of the semantic entity in the multimedia content (optional).
MediaInformation	Describes the media information that identifies the media in which the semantic entity appears (optional). MediaInformationType is defined in clause 8.
MediaInformationRef	Describes a reference to the description of media information that identifies the media in which the semantic entity appears (optional).
MediaLocator	Describes a locator that identifies the media in which the semantic entity appears (optional). This MediaLocator is an alternative to the MediaLocator in MediaInformation, and provides a more direct way of describing media locator when a full description of media information is not needed.
Mask	Describes a mask that identifies the region or segment of the multimedia content in which the semantic entity appears (optional).
AudioDescriptor	Describes audio features of the audio content at the locations pointed to by the media locator and the masks (optional).
AudioDescriptionScheme	Describes audio features of the audio content at the locations pointed to by the media locator and the masks (optional).
VisualDescriptor	Describes visual features of the visual content at the locations pointed to by the media locator and the masks (optional).
VisualDescriptionScheme	Describes visual features of the visual content at the locations pointed to by the media locator and the masks (optional).
type	Indicates the type of media occurrence. The types of media occurrences are defined as follows: • <i>perceivable</i> - The semantic entity is perceivable in the media. For example, Bill Clinton is perceivable in a picture of him.

Name	Definition
	<i>reference</i> - The semantic entity is a reference in the media. For example, Bill Clinton is a reference in a news reports about him but where he cannot be seen or heard. <i>symbol</i> - The semantic entity is symbolized in the media. For example, a picture of the Statue of Liberty symbolizes the semantic entity "freedom". This attribute value is "perceivable" by default.
Relation	Describes a relation between the semantic entity and other entities such as semantic relations, still regions, objects, events, and analytic models, among others (optional).

12.3.3 SemanticBag DS

12.3.3.1 Introduction

The SemanticBag DS extends the SemanticBase DS. The SemanticBag DS is an abstract tool that is the base of the tools that describe a collection of semantic entities and their relations.

12.3.3.2 SemanticBag DS syntax

```

<!-- ##### -->
<!-- Definition of SemanticBag DS (12.3.3) -->
<!-- ##### -->

<!-- Definition of SemanticBag DS -->
<complexType name="SemanticBagType" abstract="true">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SemanticBase" type="mpeg7:SemanticBaseType"/>
          <element name="SemanticBaseRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="Graph" type="mpeg7:GraphType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

12.3.3.3 SemanticBag DS semantics

Semantics of the SemanticBagType:

Name	Definition
SemanticBagType	This type describes a collection of semantic entities and their relations (abstract). SemanticBagType extends SemanticBaseType.
SemanticBase	Describes a semantic entity (optional).

Name	Definition
SemanticBaseRef	Describes a reference to the description of a semantic entity (optional).
Graph	Describes a graph of semantic relations among the semantic entities, including those in other narrative worlds, and possibly other non-semantic entities (optional).

12.3.4 Semantic DS

12.3.4.1 Introduction

The Semantic DS describes a narrative world that is depicted by or related to multimedia content. The SemanticRelation CS can be used to describe the relations among the semantic entities including those in other narrative worlds. The Semantic DS extends the SemanticBag DS.

12.3.4.2 Semantic DS syntax

```
<!-- ##### -->
<!-- Definition of Semantic DS (12.3.4) -->
<!-- ##### -->

<!-- Definition of Semantic DS -->
<complexType name="SemanticType">
  <complexContent>
    <extension base="mpeg7:SemanticBagType"/>
  </complexContent>
</complexType>
```

12.3.4.3 Semantic DS semantics

Semantics of the SemanticType:

Name	Definition
SemanticType	Describes a narrative world depicted by or related to multimedia content, which can be an image, a video segment, or other types of multimedia. SemanticType extends SemanticBagType.

12.3.4.4 Semantic DS examples (informative)

The following example illustrates the use of the Semantic DS for describing the semantic entity "apple". The Label elements can be used as keywords or index terms for accessing this semantic description.

```
<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics id="apple">
      <Label>
        <Name> Apple </Name>
      </Label>
      <Label>
        <Name> Fruit </Name>
      </Label>
    </Semantics>
  </Description>
</Mpeg7>
```



```

<Label>
  <Name> Pome </Name>
</Label>
<Definition>
  <FreeTextAnnotation>
    The fleshy usually rounded and red, yellow, or green edible
    pome fruit of a tree (genus Malus) of the rose family
  </FreeTextAnnotation>
</Definition>
<Property>
  <Name> Juicy </Name>
</Property>
<MediaOccurrence>
  <MediaLocator>
    <MediaUri> image.jpg </MediaUri>
  </MediaLocator>
</MediaOccurrence>
</Semantics>
</Description>
</Mpeg7>

```

12.3.5 Object DS

12.3.5.1 Introduction

The Object DS describes an object, which is a semantic entity having temporal and spatial extent. The Object DS extends the SemanticBase DS. The Object DS can be used to describe either a perceivable object or abstract object. A perceivable object is an object that exists in the physical or narrative world (e.g., "my car"). An abstract object results from the abstraction of a perceivable object (e.g. "any car"). The Object DS includes elements that describe the composition of the object from sub-objects. However, alternatively, the SemanticRelation CS can be used to describe these and other semantic relations involving the object. For example, the decomposition into sub-objects can be described using the part, component, or substance relations. The location of the object can be described using the locationOf relation. Other relations, such as the relation of the event to objects can be described using the agent and patient relations.

12.3.5.2 Object DS syntax

```

<!-- ##### -->
<!-- Definition of Object DS (12.3.5) -->
<!-- ##### -->

<!-- Definition of Object DS -->
<complexType name="ObjectType">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="Object" type="mpeg7:ObjectType"/>
        <element name="ObjectRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

12.3.5.3 Object DS semantics

Semantics of the ObjectType:

Name	Definition
ObjectType	Describes an object, which is a semantic entity having temporal and spatial extent. ObjectType extends SemanticBaseType.
Object	Describes the decomposition of the object into sub-objects by describing each sub-object (optional).
ObjectRef	Describes the decomposition of the object into sub-objects by describing a reference to each sub-object (optional).

12.3.5.4 Object DS examples (informative)

The following example illustrates the use of the Object DS for describing a table shown in an image. In this example, the table object is decomposed into one object "tabletop" and four objects "leg", i.e., a component type of decomposition. The MediaOccurrence element is used for locating the object within the media and describing the visual features of the image.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of table </Name>
      </Label>
      <SemanticBase xsi:type="ObjectType">
        <Label>
          <Name> Table </Name>
        </Label>
        <Definition>
          <FreeTextAnnotation>
            Piece of furniture consisting of a smooth flat slab
            fixed on legs
          </FreeTextAnnotation>
        </Definition>
        <MediaOccurrence>
          <MediaLocator>
            <MediaUri>image.jpg</MediaUri>
          </MediaLocator>
          <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>
        </MediaOccurrence>
        <Object id="tabletop-object">
          <Label>
            <Name> Table top </Name>
          </Label>
          <MediaOccurrence>
            <MediaLocator>
              <MediaUri>image.jpg</MediaUri>
            </MediaLocator>
            <Mask xsi:type="SpatialMaskType">
              <SubRegion>
                <Polygon>
                  <Coords mpeg7:dim="2 5"> 5 25 10 20 15 15 10 10 5 15
                  </Coords>
                </Polygon>
              </SubRegion>
            </Mask>
          </MediaOccurrence>
        </Object>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

```

        </Mask>
        <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>
    </MediaOccurrence>
</Object>
<Object id="Leg1-object">
    <Label>
        <Name> Leg </Name>
    </Label>
    <MediaOccurrence>
        <MediaLocator>
            <MediaUri>image.jpg</MediaUri>
        </MediaLocator>
        <Mask xsi:type="SpatialMaskType">
            <SubRegion>
                <Polygon>
                    <Coords mpeg7:dim="2 5"> 5 25 10 20 15 15 10 10 5 15
                </Coords>
            </Polygon>
        </SubRegion>
    </Mask>
    <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>
</MediaOccurrence>
</Object>
<Object id="Leg2-object">
    <Label>
        <Name> Leg </Name>
    </Label>
    <MediaOccurrence>
        <MediaLocator>
            <MediaUri>image.jpg</MediaUri>
        </MediaLocator>
        <Mask xsi:type="SpatialMaskType">
            <SubRegion>
                <Polygon>
                    <Coords mpeg7:dim="2 5"> 5 25 10 20 15 15 10 10 5 15
                </Coords>
            </Polygon>
        </SubRegion>
    </Mask>
    <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>
</MediaOccurrence>
</Object>
<Object id="Leg3-object">
    <Label>
        <Name> Leg </Name>
    </Label>
    <MediaOccurrence>
        <MediaLocator>
            <MediaUri>image.jpg</MediaUri>
        </MediaLocator>
        <Mask xsi:type="SpatialMaskType">
            <SubRegion>

```

```

        <Polygon>
          <Coords mpeg7:dim="2 5"> 5 25 10 20 15 15 10 10 5 15
        </Coords>
      </Polygon>
    </SubRegion>
  </Mask>
  <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>
</MediaOccurrence>
</Object>
<Object id="Leg4-object">
  <Label>
    <Name> Leg </Name>
  </Label>
  <MediaOccurrence>
    <MediaLocator>
      <MediaUri>image.jpg</MediaUri>
    </MediaLocator>
    <Mask xsi:type="SpatialMaskType">
      <SubRegion>
        <Polygon>
          <Coords mpeg7:dim="2 5"> 5 25 10 20 15 15 10 10 5 15
        </Coords>
      </Polygon>
    </SubRegion>
    </Mask>
    <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>
  </MediaOccurrence>
</Object>
</SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

12.3.6 AgentObject DS

12.3.6.1 Introduction

The AgentObject DS describes a person, organization, group of people, or personified object in a narrative world. The AgentObject DS extends the Object DS. The agent object can be decomposed into sub-objects using the object decomposition defined by the Object DS.

12.3.6.2 AgentObject DS syntax

```

<!-- ##### -->
<!-- Definition of AgentObject DS (12.3.6) -->
<!-- ##### -->

<!-- Definition of AgentObject DS -->
<complexType name="AgentObjectType">
  <complexContent>
    <extension base="mpeg7:ObjectType">
      <choice minOccurs="0">

```

```

        <element name="Agent" type="mpeg7:AgentType"/>
        <element name="AgentRef" type="mpeg7:ReferenceType"/>
    </choice>
</extension>
</complexContent>
</complexType>

```

12.3.6.3 AgentObject DS semantics

Semantics of the AgentObjectType:

Name	Definition
AgentObjectType	Describes a person, organization, group of people, or personified object in a narrative world. AgentObjectType extends ObjectType.
Agent	Describes the agent represented by the object (optional).
AgentRef	Describes a reference to the agent represented by the object (optional).

12.3.6.4 AgentObject DS examples (informative)

The following example illustrates the use of the AgentObject DS for describing a person named Alexander (Alex) Green.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of person </Name>
      </Label>
      <SemanticBase xsi:type="AgentObjectType">
        <Label>
          <Name> Student </Name>
        </Label>
        <Definition>
          <FreeTextAnnotation> Student named Alexander Green
          </FreeTextAnnotation>
        </Definition>
        <Agent xsi:type="PersonType">
          <Name>
            <GivenName abbrev="Alex"> Alexander </GivenName>
            <FamilyName> Green </FamilyName>
          </Name>
        </Agent>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the AgentObject DS for describing members of a baseball team. In this example, the agent object (Baseball team) is composed of three agent objects (players – pitcher, catcher, and third base).

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">

```

```

<Semantics>
  <Label>
    <Name> Baseball team </Name>
  </Label>
  <SemanticBase xsi:type="AgentObjectType">
    <Label>
      <Name> NY Mets </Name>
    </Label>
    <Object xsi:type="AgentObjectType" id="player1-agent">
      <Label href="urn:example:baseball">
        <Name> Pitcher </Name>
      </Label>
      <Agent xsi:type="PersonType">
        <Name>
          <GivenName> Al </GivenName>
          <FamilyName> Leiter </FamilyName>
        </Name>
      </Agent>
    </Object>
    <Object xsi:type="AgentObjectType" id="player2-agent">
      <Label href="urn:example:baseball">
        <Name> Catcher </Name>
      </Label>
      <Agent xsi:type="PersonType">
        <Name>
          <GivenName> Mike </GivenName>
          <FamilyName> Piazza </FamilyName>
        </Name>
      </Agent>
    </Object>
    <Object xsi:type="AgentObjectType" id="player3-agent">
      <Label href="urn:example:baseball">
        <Name> Third base </Name>
      </Label>
      <Agent xsi:type="PersonType">
        <Name>
          <GivenName> Robin </GivenName>
          <FamilyName> Ventura </FamilyName>
        </Name>
      </Agent>
    </Object>
  </SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

12.3.7 Event DS

12.3.7.1 Introduction

The Event DS describes an event, which is a semantic activity that takes place at a particular time or in a particular location. The Event DS extends the SemanticBase DS. The Event DS can describe either a perceivable or an abstract event in a narrative world. A perceivable event is a dynamic relation involving one or more objects taking place in time and space in a narrative world (e.g., "Alex shaking hands with Ana"). An abstract event results from abstraction of a perceivable event (e.g., "Alex shaking hands with any woman").

The Event DS includes elements that describe the composition of the event from sub-events, in addition to the location and time of the event. However, alternatively, the SemanticRelation CS can be used to describe these and other semantic relations involving the event. For example, the decomposition into sub-events can be described using the part relation. The location of the event can be described using the location relation. The time of the

event can be described using the time relation. Other relations, such as the relation of the event to objects can be described using the agent and patient relations.

12.3.7.2 Event DS syntax

```
<!-- ##### -->
<!-- Definition of Event DS (12.3.7) -->
<!-- ##### -->

<!-- Definition of Event DS -->
<complexType name="EventType">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Event" type="mpeg7:EventType"/>
          <element name="EventRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="SemanticPlace" type="mpeg7:SemanticPlaceType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="SemanticTime" type="mpeg7:SemanticTimeType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

12.3.7.3 Event DS semantics

Semantics of the EventType:

Name	Definition
EventType	Describes an event that takes place at a particular time or in a particular location. EventType extends SemanticBaseType.
Event	Describes the decomposition of the event into sub-events by describing each sub-event (optional).
EventRef	Describes the decomposition of the event into sub-events by describing a reference to each sub-event (optional).
SemanticPlace	Describes the location or spatial extent of the event (optional).
SemanticTime	Describes a time or temporal duration of the event (optional).

12.3.7.4 Event DS examples (informative)

The following example illustrates the use of the Event DS for describing the event of a "handshake between two colleagues". In this example, the Relation DS uses the SemanticRelation CS to describe the agent and accompanier of the action.

```
<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
```

```

<Label>
  <Name> Shake hands </Name>
</Label>
<SemanticBase xsi:type="AgentObjectType" id="A0a">
  <Label href="urn:example:acs">
    <Name> Person A </Name>
  </Label>
</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="A0b">
  <Label href="urn:example:acs">
    <Name> Person B </Name>
  </Label>
</SemanticBase>
<SemanticBase xsi:type="EventType" id="EV1">
  <Label href="urn:example:acs">
    <Name> Shake hands </Name>
  </Label>
  <Definition>
    <FreeTextAnnotation>
      Clasping of right hands by two people.
    </FreeTextAnnotation>
  </Definition>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
    target="#A0a"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:accompanier"
    target="#A0b"/>
</SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

12.3.8 Concept DS

12.3.8.1 Introduction

The Concept DS describes a semantic concept, which is a semantic entity that cannot be described as an abstraction of the other semantic entities such as object or event. The Concept DS extends the SemanticBase DS. Example concepts include notions such as "harmony," "full-bodied," or "freedom". A concept may involve multiple objects and events, metaphors, analogies, and blending together of other concepts. The Concept DS describes concepts as collection of properties.

12.3.8.2 Concept DS syntax

```

<!-- ##### -->
<!-- Definition of Concept DS (11.3.8) -->
<!-- ##### -->

<!-- Definition of Concept DS -->
<complexType name="ConceptType">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType"/>
  </complexContent>
</complexType>

```


12.3.8.3 Concept DS semantics

Semantics of the ConceptType:

Name	Definition
ConceptType	Describes a semantic concept, which is a semantic entity that cannot be described as an abstraction of the other semantic entities such as object or event. ConceptType extends SemanticBaseType.

12.3.8.4 Concept DS examples (informative)

The following example illustrates the use of the Concept DS for describing the concept "freedom" as having properties of "open," "outspoken," and "frank". In this example, the description identifies an image of the Statue of Liberty that symbolizes "freedom".

```
<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Concept of freedom </Name>
      </Label>
      <SemanticBase xsi:type="ConceptType" id="freedom">
        <Label>
          <Name> Freedom </Name>
        </Label>
        <Property>
          <Name> Open </Name>
        </Property>
        <Property>
          <Name> Outspoken </Name>
        </Property>
        <Property>
          <Name> Frank </Name>
        </Property>
        <MediaOccurrence type="symbol">
          <MediaLocator>
            <MediaUri> liberty.gif
          </MediaUri>
          </MediaLocator>
        </MediaOccurrence>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>
```

12.3.9 SemanticState DS

12.3.9.1 Introduction

The SemanticState DS extends the SemanticBase DS. The SemanticState DS describes a state or parametric attributes of a semantic entity or semantic relation, at a given time or location in a narrative world, or in a given media location (e.g., height and weight). A state is a set of numerical and verbal attributes that can be attached to semantic entities such as objects and events and other semantic elements such as semantic relations.

12.3.9.2 SemanticState DS syntax

```

<!-- ##### -->
<!-- Definition of SemanticState DS (12.3.9) -->
<!-- ##### -->

<!-- Definition of SemanticState DS -->
<complexType name="SemanticStateType">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <sequence>
        <element name="AttributeValuePair"
          minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="Attribute" type="mpeg7:TermUseType"/>
              <element name="Unit" type="mpeg7:TermUseType"
                minOccurs="0"/>
              <choice minOccurs="1" maxOccurs="unbounded">
                <element name="BooleanValue" type="boolean"/>
                <element name="IntegerValue" type="integer"/>
                <element name="FloatValue" type="float"/>
                <element name="IntegerMatrixValue"
                  type="mpeg7:IntegerMatrixType"/>
                <element name="FloatMatrixValue"
                  type="mpeg7:FloatMatrixType"/>
                <element name="TextValue"
                  type="mpeg7:TextualType"/>
                <element name="TextAnnotationValue"
                  type="mpeg7:TextAnnotationType"/>
                <element name="ControlledTermUseValue"
                  type="mpeg7:ControlledTermUseType"/>
                <element name="DescriptorValue"
                  type="mpeg7:DType"/>
              </choice>
            </sequence>
          </complexType>
        </element>
        <element name="SemanticTime" type="mpeg7:SemanticTimeType"
          minOccurs="0"/>
        <element name="SemanticPlace" type="mpeg7:SemanticPlaceType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

12.3.9.3 SemanticState DS semantics

Name	Definition
SemanticStateType	Describes parametric attributes of a semantic entity or a semantic relation at a given time or location in the narrative world, or in a given media location. SemanticStateType extends SemanticBaseType.
AttributeValuePair	Describes one pair formed by an attribute and a value, the value being a combination of a series of types (e.g. Boolean, integer, real, matrix, string, and controlled term). The Unit element allows the specification of the units used for the parameter value.
Attribute	Indicates the name of the attribute
Unit	Indicates the unit of the attribute value (optional).
BooleanValue	Indicates part of the value of the attribute as a Boolean.
IntegerValue	Indicates part of the value of the attribute as an integer.
FloatValue	Indicates part of the value of the attribute as a float.
IntegerMatrixValue	Indicates part of the value of the attribute as a matrix of integers.
FloatMatrixValue	Indicates part of the value of the attribute as a matrix of floats.
TextValue	Indicates part of the value of the attribute as a text string.
TextAnnotationValue	Indicates part of the value of the attribute as a text annotation.
ControlledTermUseValue	Indicates part of the value of the attribute as a controlled term.
DescriptorValue	Describes part of the value of the attribute as a descriptor.
SemanticTime	Describes semantically the time associated with the state (optional).
SemanticPlace	Describes semantically the location associated with the state (optional).

12.3.9.4 SemanticState DS examples (informative)

The following example illustrates the use of the SemanticState DS for describing semantic attributes of a sunset. In this example, the event "sunset-event" represents the event "sunset" and the semantic state "sunset-state" describes the semantic attributes of the event "sunset". A graph relates the event "sunset-event" and the semantic state "sunset-state". A sunset is considered as an event because it can be nominalized: "The sunset is beautiful".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Sunset </Name>
      </Label>
      <SemanticBase xsi:type="EventType" id="sunset-event">
        <Label>
          <Name> Sunset </Name>
        </Label>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

```

    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="SemanticStateType" id="sunset-state">
    <Label>
      <Name> State of sunset </Name>
    </Label>
    <AttributeValuePair>
      <Attribute>
        <Name> Blue </Name>
      </Attribute>
      <IntegerValue>90</IntegerValue>
    </AttributeValuePair>
    <AttributeValuePair>
      <Attribute>
        <Name> Pink </Name>
      </Attribute>
      <IntegerValue>9</IntegerValue>
    </AttributeValuePair>
    <AttributeValuePair>
      <Attribute>
        <Name> Orange </Name>
      </Attribute>
      <IntegerValue>50</IntegerValue>
    </AttributeValuePair>
    <AttributeValuePair>
      <Attribute>
        <Name> Yellow </Name>
      </Attribute>
      <IntegerValue>45</IntegerValue>
    </AttributeValuePair>
  </SemanticBase>
  <Graph>
    <!-- Relate event sunset-event and state sunset-state -->
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:state"
      source="#sunset-event" target="#sunset-state"/>
  </Graph>
</Semantics>
</Description>
</Mpeg7>

```

The following example illustrates the use of the SemanticState DS for describing the place and event for a person. In this example, "John is in Singapore today" is described by linking the AgentObject DS description corresponding to "John," the SemanticPlace DS description of "Singapore," and the SemanticTime DS description of "today" using the state "position-state". This description may be represented by an event "Being" taking place during semantic time "today" in semantic place "Singapore".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Place and time for a person </Name>
      </Label>
      <SemanticBase xsi:type="AgentObjectType" id="agent-person">
        <Label>
          <Name> John </Name>
        </Label>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:state"
          target="#position-state"/>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

```

<SemanticBase xsi:type="SemanticStateType" id="position-state">
  <Label>
    <Name> Position </Name>
  </Label>
  <AttributeValuePair>
    <Attribute>
      <Name> Position </Name>
    </Attribute>
    <IntegerValue>1</IntegerValue>
  </AttributeValuePair>
  <SemanticTime>
    <Label>
      <Name> today </Name>
    </Label>
  </SemanticTime>
  <SemanticPlace>
    <Label>
      <Name> Singapore </Name>
    </Label>
  </SemanticPlace>
</SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

12.3.10 SemanticPlace DS

12.3.10.1 Introduction

The SemanticPlace DS extends the SemanticBase DS. The SemanticPlace DS describes location in a narrative world. The SemanticPlace DS is a specialized SemanticBase DS that encapsulates the Place DS (see clause 7) and semantic place interval information within the SemanticBase DS.

12.3.10.2 SemanticPlace DS syntax

```

<!-- ##### -->
<!-- Definition of SemanticPlace DS (12.3.10) -->
<!-- ##### -->

<!-- Definition of SemanticPlace DS -->
<complexType name="SemanticPlaceType">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <sequence>
        <element name="Place" type="mpeg7:PlaceType" minOccurs="0"/>
        <element name="SemanticPlaceInterval"
          minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="Position" type="mpeg7:PositionType"
                minOccurs="0"/>
              <element name="Extent" type="mpeg7:ExtentType"
                minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```
</complexContent>
</complexType>
```

12.3.10.3 SemanticPlace DS semantics

Semantics of the SemanticPlaceType:

Name	Definition
SemanticPlaceType	Describes location in a narrative world as a place or the spatial interval related to a place. SemanticPlaceType extends SemanticBaseType.
Place	Describes the place represented by the location (optional).
SemanticPlaceInterval	Describes the position or the spatial extent of an interval of the place (optional).
Position	Semantically describes the position of the place interval as a point in space (optional).
Extent	Semantically describes the spatial extent of the place interval in relation to the position, for instance, distance or area from or around the position, respectively (optional).

12.3.10.4 SemanticPlace DS examples (informative)

The following example illustrates the use of the SemanticPlace DS for describing the location "Columbia University in the City of New York".

```
<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Semantic place for Columbia University </Name>
      </Label>
      <SemanticBase xsi:type="SemanticPlaceType" id="columbia-location">
        <Label>
          <Name> Columbia University </Name>
        </Label>
        <Place>
          <Name xml:lang="en">
            Columbia University in the City of New York </Name>
          <Region> us </Region>
          <PostalAddress>
            <AddressLine> 600 West 116th Street, New York, NY</AddressLine>
            <PostingIdentifier> U-10027 </PostingIdentifier>
          </PostalAddress>
        </Place>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>
```

The following example illustrates the use of the SemanticPlaceInterval element in SemanticPlace DS for describing the location "4 miles".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Semantic place for location four miles </Name>
      </Label>
      <SemanticBase xsi:type="SemanticPlaceType" id="miles4-place">
        <Label>
          <Name> 4 miles </Name>
        </Label>
        <SemanticPlaceInterval>
          <Extent measurementType="length" unit="mile" value="4"/>
        </SemanticPlaceInterval>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the `SemanticPlaceInterval` element in `SemanticPlace` DS for describing the location "Within 4 miles radius of New York City".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Semantic place for four mile radius of NYC </Name>
      </Label>
      <SemanticBase xsi:type="SemanticPlaceType" id="miles4NYC-place">
        <Label>
          <Name> Within 4 miles radius of New York City </Name>
        </Label>
        <SemanticPlaceInterval>
          <Position origin="New York City"/>
          <Extent measurementType="area" unit="square miles" value="16*pi"/>
        </SemanticPlaceInterval>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the `SemanticPlaceInterval` element in `SemanticPlace` DS for describing the location "5 kilometers north of New York City and 40 kilometers east of Boston".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Semantic place for location near NYC and Boston </Name>
      </Label>
      <SemanticBase xsi:type="SemanticPlaceType" id="NYCBoston-place">
        <Label>
          <Name>
            5 kilometers north of New York City and
            40 kilometers east of Boston
          </Name>
        </Label>
        <SemanticPlaceInterval>
          <Position origin="New York City">

```

```

        <Displacement measurementType="length" unit="kilometers"
            value="5"/>
        <Direction measurementType="direction" unit="direction"
            value="north"/>
    </Position>
</SemanticPlaceInterval>
<SemanticPlaceInterval>
    <Position origin="Boston">
        <Displacement measurementType="length" unit="kilometers"
            value="40"/>
        <Direction measurementType="direction" unit="direction"
            value="east"/>
    </Position>
</SemanticPlaceInterval>
</SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

If a semantic place is complex, relations may relate the location to the other semantic entities such as objects and events needed to fully describe it. The following example illustrates the use of the SemanticPlace DS for describing the location "the place where Cri and Ale got engaged."

In this example, the location "at the Eiffel Tower in the same spot where Cri and Ale were engaged last year" and the event "Spot at the Eiffel Tower" are described in the semantic entities "sem2-loc" and "engagement-event", respectively. The new semantic place ("sem2-loc") can be defined as the same as the SemanticPlace DS description of the event "engagement-event" – relation identifier to semantic place "Eiffel-loc" (possibility one) – or as being the location where that event took place – relation locationOf to the event "engagement-event" (possibility two). In general, if the same semantic entity is described by multiple semantic descriptions, the relationship of the descriptions can be described using the semantic relation identifier.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> The place where Cri and Ale got engaged </Name>
      </Label>
      <SemanticBase xsi:type="SemanticPlaceType" id="sem2-loc">
        <Label>
          <Name> Eiffel Tower </Name>
        </Label>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:identifier"
            target="#Eiffel-loc"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:locationOf"
            target="#engagement-event"/>
      </SemanticBase>
      <SemanticBase xsi:type="EventType" id="engagement-event">
        <Label>
          <Name> Engagement </Name>
        </Label>
        <Definition>
          <FreeTextAnnotation>
            Engagement of Cri and Ale, at the Eiffel Tower in Paris
          </FreeTextAnnotation>
        </Definition>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
            target="#Eiffel-loc"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
            target="#lastyear-time"/>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```



```

    <SemanticBase xsi:type="SemanticPlaceType" id="Eiffel-loc">
      <Label>
        <Name> Spot at the Eiffel Tower </Name>
      </Label>
    </SemanticBase>
    <SemanticBase xsi:type="SemanticTimeType" id="lastyear-time">
      <Label>
        <Name> Last year </Name>
      </Label>
      <SemanticTimeInterval>
        <TimePoint origin="now">
          <Displacement measurementType="length" unit="year" value="1"/>
          <Direction measurementType="direction" unit="direction"
            value="before"/>
        </TimePoint>
      </SemanticTimeInterval>
    </SemanticBase>
  </Semantics>
</Description>
</Mpeg7>

```

12.3.11 SemanticTime DS

12.3.11.1 Introduction

The SemanticTime DS extends the SemanticBase DS. The SemanticTime DS describes time in a narrative world. The SemanticTime DS is a specialized SemanticBase DS that encapsulates the Time datatype (see clause 6) and semantic relative time information within the SemanticBase DS.

12.3.11.2 SemanticTime DS syntax

```

<!-- ##### -->
<!-- Definition of SemanticTime DS (12.3.11) -->
<!-- ##### -->

<!-- Definition of SemanticTime DS -->
<complexType name="SemanticTimeType">
  <complexContent>
    <extension base="mpeg7:SemanticBaseType">
      <sequence>
        <element name="Time" type="mpeg7:TimeType" minOccurs="0"/>
        <element name="SemanticTimeInterval"
          minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="TimePoint" type="mpeg7:PositionType"
                minOccurs="0"/>
              <element name="Duration" type="mpeg7:ExtentType"
                minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

12.3.11.3 SemanticTime DS semantics

Semantics of the SemanticTimeType:

Name	Definition
SemanticTimeType	Describes time in a narrative world as a time point or a time interval. SemanticTimeType extends SemanticBaseType.
Time	Describes the time point and the duration with numerical attributes of the time (optional).
SemanticTimeInterval	Describes the time point or the duration of an interval of the time (optional).
TimePoint	Semantically describes the time point of the time interval (optional). The Direction element shall indicate what direction to measure the time in from the origin, for instance, "before" or "after".
Duration	Semantically describes the duration of the time interval (optional).

12.3.11.4 SemanticTime DS examples (informative)

The following example illustrates the use of the Time DS in SemanticTime DS for describing the time "10 days starting at 14:13:00 hours, 12 November, 2001".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> 10 days </Name>
      </Label>
      <SemanticBase xsi:type="SemanticTimeType">
        <Label>
          <Name>
            10 days starting at 14:13:00 hours, 12 November, 2001
          </Name>
        </Label>
        <Time>
          <TimePoint>2001-11-12T14:13+01:00</TimePoint>
          <Duration>P10D</Duration>
        </Time>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the SemanticTimeInterval element in SemanticTime DS for describing the time "4 weeks".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of 4 weeks </Name>
      </Label>

```

```

        <SemanticBase xsi:type="SemanticTimeType">
            <Label>
                <Name> 4 weeks </Name>
            </Label>
            <SemanticTimeInterval>
                <Duration measurementType="length" unit="week" value="4"/>
            </SemanticTimeInterval>
        </SemanticBase>
    </Semantics>
</Description>
</Mpeg7>

```

The following example illustrates the use of the `SemanticTimeInterval` element in `SemanticTime` DS for describing the time "during 1997".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of 1997 </Name>
      </Label>
      <SemanticBase xsi:type="SemanticTimeType">
        <Label>
          <Name> During 1997 </Name>
        </Label>
        <SemanticTimeInterval>
          <TimePoint origin="January 1, 1997">
            <Displacement measurementType="length" unit="year" value="1"/>
            <Direction measurementType="during" unit="direction"
              value="after"/>
          </TimePoint>
        </SemanticTimeInterval>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the `SemanticTimeInterval` element in `SemanticTime` DS for describing the time "last year".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of last year </Name>
      </Label>
      <SemanticBase xsi:type="SemanticTimeType">
        <Label>
          <Name> Last year </Name>
        </Label>
        <SemanticTimeInterval>
          <TimePoint origin="now">
            <Displacement measurementType="length" unit="year" value="1"/>
            <Direction measurementType="direction" unit="direction"
              value="before"/>
          </TimePoint>
        </SemanticTimeInterval>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

```

</Description>
</Mpeg7>

```

The following example illustrates the use of the `SemanticTimeInterval` element in `SemanticTime` DS for describing the time "the third and fourth day in April".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of April 3 to 4 </Name>
      </Label>
      <SemanticBase xsi:type="SemanticTimeType">
        <Label>
          <Name> The third and fourth day in April </Name>
        </Label>
        <SemanticTimeInterval>
          <TimePoint origin="April">
            <Displacement measurementType="length" unit="day" value="3"/>
            <Direction measurementType="direction" unit="direction"
              value="after"/>
          </TimePoint>
          <Duration measurementType="length" unit="day" value="2"/>
        </SemanticTimeInterval>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the `SemanticTimeInterval` element in `SemanticTime` DS for describing the time "Monday and Friday from 9am to 5pm".

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of the time Monday and Friday 9-5 </Name>
      </Label>
      <SemanticBase xsi:type="SemanticTimeType">
        <Label>
          <Name> Monday and Friday from 9am to 5pm </Name>
        </Label>
        <SemanticTimeInterval>
          <TimePoint origin="Monday">
            <Displacement measurementType="length" unit="hour" value="9"/>
            <Direction measurementType="direction" unit="direction"
              value="after"/>
          </TimePoint>
          <Duration measurementType="length" unit="hour" value="8"/>
        </SemanticTimeInterval>
        <SemanticTimeInterval>
          <TimePoint origin="Friday">
            <Displacement measurementType="length" unit="hour" value="9"/>
            <Direction measurementType="direction" unit="direction"
              value="after"/>
          </TimePoint>
          <Duration measurementType="length" unit="hour" value="8"/>
        </SemanticTimeInterval>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

```

    </Semantics>
  </Description>
</Mpeg7>

```

If a semantic time is complex, relations may relate the time to the semantic entities such as objects and events needed to fully describe it. The following example illustrates the use of the SemanticTime DS for describing the time "in the year 1899 at the same time when the earthquake happened in San Francisco".

In this example, the time "in the year 1899 at the same time when the earthquake happened in San Francisco" and the event "Earthquake in San Francisco in 1899" are described in the semantic entities "sem3-time" and "earthquake-event", respectively. The new semantic time ("sem3-time") can be defined as the same as the SemanticTime DS description of the event "earthquake-event" – relation identifier to semantic time "m3_1899-time" (possibility one) - or as being the time when that event happened –relation timeOf to the event "earthquake-event" (possibility two).

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Description of time of the earthquake </Name>
      </Label>
      <SemanticBase xsi:type="SemanticTimeType" id="sem3-time">
        <Label>
          <Name> Time </Name>
        </Label>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:identifier"
          target="#m3_1899-time"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:timeOf"
          target="#earthquake-event"/>
      </SemanticBase>
      <SemanticBase xsi:type="EventType" id="earthquake-event">
        <Label>
          <Name> Earthquake </Name>
        </Label>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
          target="#m3_1899-time"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
          target="#sanfran-loc"/>
      </SemanticBase>
      <SemanticBase xsi:type="SemanticTimeType" id="m3_1899-time">
        <Label>
          <Name> 12 November, 1899 </Name>
        </Label>
        <Time>
          <TimePoint>1899-11-12T14:13:00</TimePoint>
          <Duration>PT3M</Duration>
        </Time>
      </SemanticBase>
      <SemanticBase xsi:type="SemanticPlaceType" id="sanfran-loc">
        <Label>
          <Name> San Francisco </Name>
        </Label>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

12.3.12 Semantic entity description tools examples (informative)

The following example illustrates the use of the Semantic DS and other SemanticBase DSs for describing the narrative world of an image as shown in Figure 55. In this example, the AgentObject DS, the Event DS, the SemanticPlace DS, the SemanticTime DS, and the Concept DS describe the people, the event, the place, the time and a concept depicted or symbolized in the image. The semantic description of the image is the following: "Alex is shaking hands with Ana in New York during 1997. The event is showing as comradeship". The example assumes the image and the regions of the two persons have been described using the StillRegion DS and the StillRegionSpatialDecomposition DS.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image id="SR1">
        <SpatialDecomposition gap="true" overlap="false">
          <StillRegion id="SR2">
            <TextAnnotation>
              <FreeTextAnnotation> Alex </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>
          </StillRegion>
          <StillRegion id="SR3">
            <TextAnnotation>
              <FreeTextAnnotation> Ana </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>
          </StillRegion>
        </SpatialDecomposition>
      </Image>
    </MultimediaContent>
  </Description>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name>Semantic description of the handshake </Name>
      </Label>
      <SemanticBase xsi:type="EventType" id="EV1">
        <Label>
          <Name>Shake hands</Name>
        </Label>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
          target="#A01"/>
        <Relation
          type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:accompanier"
          target="#A02"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:property"
          target="#C1"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
          target="#SR1"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
          target="#SP1"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
          target="#ST1"/>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

```

<SemanticBase xsi:type="AgentObjectType" id="A01">
  <Label>
    <Name>Alex</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
    target="#SR2"/>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Alex</GivenName>
    </Name>
  </Agent>
</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="A02">
  <Label>
    <Name>Ana</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depictedBy"
    target="#SR3"/>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Ana</GivenName>
    </Name>
  </Agent>
</SemanticBase>
<SemanticBase xsi:type="ConceptType" id="C1">
  <Label>
    <Name> Comradeship</Name>
  </Label>
  <Property>
    <Name> Partnership</Name>
  </Property>
  <Property>
    <Name> Friendship</Name>
  </Property>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:symbolizedBy"
    target="#SR1"/>
</SemanticBase>
<SemanticBase xsi:type="SemanticPlaceType" id="SP1">
  <Label>
    <Name>New York</Name>
  </Label>
  <Place>
    <!-- more elements here -->
  </Place>
</SemanticBase>
<SemanticBase xsi:type="SemanticTimeType" id="ST1">
  <Label>
    <Name> During 1997 </Name>
  </Label>
  <SemanticTimeInterval>
    <TimePoint origin="January 1, 1997">
      <Displacement measurementType="length" unit="year" value="1"/>
      <Direction measurementType="during" unit="direction"
        value="after"/>
    </TimePoint>
  </SemanticTimeInterval>
</SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

12.4 Semantic attribute description tools

12.4.1 Introduction

This subclause specifies tools for describing attributes of semantic entities related to abstraction in semantic descriptions and semantic measurements in time and space. The following table summarizes the functionality of each semantic attribute and its application to the different types of semantic entities.

Table 32 — Overview of the semantic attribute description tools

<i>Tool</i>	<i>Functionality</i>	<i>Entities</i>
AbstractionLevel datatype	This tool describes the abstraction performed in the description of a semantic entity.	SemanticBase DS
Extent datatype	This tool describes the extent or size in semantic terms.	SemanticPlace DS SemanticTime DS
Position datatype	This tool describes the position in semantic terms.	SemanticPlace DS SemanticTime DS

12.4.2 AbstractionLevel datatype

12.4.2.1 Introduction

The AbstractionLevel datatype describes the kind of abstraction that has been performed in a description of a semantic entity (see 12.2). The AbstractionLevel datatype applies to the SemanticBase DS. When it is not present in the description of a semantic entity, then the description is concrete – it describes the narrative world depicted by multimedia content and references the multimedia content (e.g., through the MediaOccurrence element in SemanticBase DS or using relations from SemanticRelation CS). If the AbstractionLevel is present in the description, the description is a media abstraction or a formal abstraction.

12.4.2.2 AbstractionLevel datatype syntax

```

<!-- ##### -->
<!-- Definition of AbstractionLevel datatype (12.4.2) -->
<!-- ##### -->

<!-- Definition of AbstractionLevel DS -->
<complexType name="AbstractionLevelType">
  <attribute name="dimension" type="nonNegativeInteger" use="required"/>
</complexType>

```

12.4.2.3 AbstractionLevel datatype semantics

Semantics of the AbstractionLevelType:

<i>Name</i>	<i>Definition</i>
AbstractionLevelType	Describes the abstraction performed in the description of a semantic entity.

Name	Definition
dimension	Indicates the level of abstraction of the description. The levels of abstraction are defined as follows: • A value of 0 indicates a media abstraction. A media abstraction is a semantic description that has been separated from the multimedia content, and describes all instances of multimedia content that are similar. • A value of 1 indicates a formal abstraction. A formal abstraction is a semantic description that contains semantic entities acting as variables common to a set of semantic descriptions. • Higher values indicate an abstraction of formal abstractions. In this abstraction, the relevant parts of the description are expressed as graphs.

The usage of AbstractionLevel datatype can be summarized as follows: the abstraction level is not used when describing the semantics of a particular instance of multimedia content. In this case, the semantic description is not a class for accessing or indexing multimedia content, it simply describes what is shown in the multimedia content.

The abstraction level can be used when separated from the multimedia content to describe all those instances of multimedia content which share the semantic description. In this case, an abstraction level of 0 indicates this as a *media abstraction*. If one or more (or even all) the specific objects, events or other semantic entities are replaced by classes, the semantic description is a *formal abstraction*, and has an abstraction level of 1. Values higher than one indicates a formal abstraction of the formal abstraction. Values higher than 2 may be used but are increasingly specialized and less common.

12.4.2.4 AbstractionLevel datatype examples (informative)

The following example illustrates the use of the AbstractionLevel datatype in the SemanticBase DS for describing a media abstraction of the concrete semantic description shown in Figure 55 and discussed in 12.3.12, which is illustrated in Figure 56(a). In this example, the media abstraction is the description "Alex is shaking hands with Ana in NewYork on the 9th of September" with no links to the image; this description is, therefore, applicable to any image or video depicting the event.

```

<!-- Media abstraction: AbstractionLevel dimension = 0 -->
<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <AbstractionLevel dimension="0"/>
      <Label>
        <Name>Alex shakes hands with Ana </Name>
      </Label>
      <SemanticBase xsi:type="EventType" id="EV1-MA">
        <Label>
          <Name>Shake hands</Name>
        </Label>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
          target="#A01-MA"/>
        <Relation
          type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:accompanier"
          target="#A02-MA"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:property"
          target="#C1-MA"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
          target="#SP1-MA"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
          target="#ST1-MA"/>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

```

</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="A01-MA">
  <Label>
    <Name>Alex</Name>
  </Label>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Alex</GivenName>
    </Name>
  </Agent>
</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="A02-MA">
  <Label>
    <Name>Ana</Name>
  </Label>
  <Agent xsi:type="PersonType">
    <Name>
      <GivenName>Ana</GivenName>
    </Name>
  </Agent>
</SemanticBase>
<SemanticBase xsi:type="ConceptType" id="C1-MA">
  <Label>
    <Name> Comradeship</Name>
  </Label>
  <Property>
    <Name> Friendship</Name>
  </Property>
</SemanticBase>
<SemanticBase xsi:type="SemanticPlaceType" id="SP1-MA">
  <Label>
    <Name>Columbia University</Name>
  </Label>
</SemanticBase>
<SemanticBase xsi:type="SemanticTimeType" id="ST1-MA">
  <Label>
    <Name>September 9</Name>
  </Label>
</SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

The following example illustrates the use of the `AbstractionLevel` datatype in the `SemanticBase` DS for describing a formal abstraction of the concrete semantic description shown in Figure 55 and discussed in 12.3.12, which is illustrated in Figure 56.b. In this example, the formal abstraction is the description "Alex is shaking hands with any woman in New York on the 9th of September," where "any woman" is the variable of the formal abstraction.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <AbstractionLevel dimension="1"/>
      <Label>
        <Name>Alex shakes hands with any woman </Name>
      </Label>
      <SemanticBase xsi:type="EventType" id="EV1-FA">
        <Label>
          <Name>Shake hands</Name>
        </Label>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"

```

```

        target="#A01-FA"/>
    <Relation
        type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:accompanier"
        target="#A02-FA"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:property"
        target="#C1-FA"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:location"
        target="#SP1-FA"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:time"
        target="#ST1-FA"/>
</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="A01-FA">
    <Label>
        <Name>Alex</Name>
    </Label>
    <Agent xsi:type="PersonType">
        <Name>
            <GivenName>Alex</GivenName>
        </Name>
    </Agent>
</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="A02-FA">
    <AbstractionLevel dimension="1"/>
    <Label>
        <Name>Any woman</Name>
    </Label>
</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="A01-MA">
    <Label>
        <Name>Alex</Name>
    </Label>
    <Agent xsi:type="PersonType">
        <Name>
            <GivenName>Alex</GivenName>
        </Name>
    </Agent>
</SemanticBase>
<SemanticBase xsi:type="AgentObjectType" id="A02-MA">
    <Label>
        <Name>Ana</Name>
    </Label>
    <Agent xsi:type="PersonType">
        <Name>
            <GivenName>Ana</GivenName>
        </Name>
    </Agent>
</SemanticBase>
<SemanticBase xsi:type="ConceptType" id="C1-MA">
    <Label>
        <Name> Comradeship</Name>
    </Label>
    <Property>
        <Name> Friendship</Name>
    </Property>
</SemanticBase>
<SemanticBase xsi:type="SemanticPlaceType" id="SP1-MA">
    <Label>
        <Name>Columbia University</Name>
    </Label>
</SemanticBase>
<SemanticBase xsi:type="SemanticTimeType" id="ST1-MA">

```

```

        <Label>
          <Name>September 9</Name>
        </Label>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the SemanticRelation CS for creating a concrete semantic description from the formal abstraction above. In this example, the variable "any woman" in the formal abstraction is filled in with the concrete instance "Joana" using the semantic relation exemplifies.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Alex shaking hands with Joana </Name>
      </Label>
      <MediaOccurrence>
        <MediaLocator>
          <MediaUri>content.mpg</MediaUri>
        </MediaLocator>
      </MediaOccurrence>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"
        target="#SM1-FA"/>
      <SemanticBase xsi:type="AgentObjectType" id="A03">
        <Label>
          <Name> Joana </Name>
        </Label>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"
          target="#A02-FA"/>
        <Agent xsi:type="PersonType">
          <Name>
            <GivenName>Joana</GivenName>
          </Name>
        </Agent>
      </SemanticBase>
    </Semantics>
  </Description>
</Mpeg7>

```

12.4.3 Extent datatype

12.4.3.1 Introduction

The Extent datatype describes the size or extent in semantic terms. The Extent datatype applies to the SemanticPlace DS and the SemanticTime DS.

12.4.3.2 Extent datatype syntax

```

<!-- ##### -->
<!-- Definition of Extent datatype (12.4.3) -->
<!-- ##### -->

<!-- Definition of Extent datatype -->
<complexType name="ExtentType">
  <attribute name="measurementType" type="string" use="required"/>

```

```

<attribute name="unit" type="string" use="required"/>
<attribute name="value" type="string" use="required"/>
</complexType>

```

12.4.3.3 Extent datatype semantics

Semantics of the ExtentType:

<i>Name</i>	<i>Definition</i>
ExtentType	Describes the size or the extent with respect to a measurement type.
measurementType	Indicates the type of extent. Examples of measurement types are "length," "weight," and "temperature".
unit	Indicates the unit of the value attribute.
value	Indicates the value of the extent.

12.4.4 Position datatype

12.4.4.1 Introduction

The Position datatype describes the position in semantic terms. The Position datatype applies to the SemanticPlace DS and the SemanticTime DS.

12.4.4.2 Position datatype syntax

```

<!-- ##### -->
<!-- Definition of Position datatype (12.4.4) -->
<!-- ##### -->

<!-- Definition of Position datatype -->
<complexType name="PositionType">
  <sequence>
    <element name="Displacement" type="mpeg7:ExtentType" minOccurs="0"/>
    <element name="Direction" type="mpeg7:ExtentType" minOccurs="0"/>
  </sequence>
  <attribute name="origin" type="string" use="optional"/>
</complexType>

```

12.4.4.3 Position datatype semantics

Semantics of the `PositionType`:

<i>Name</i>	<i>Definition</i>
<code>PositionType</code>	Describes the position, for example, time point, location, or altitude.
<code>Displacement</code>	Describes the distance from the origin of the position (optional).
<code>Direction</code>	Describes the direction or the angular displacement from the origin of the position (optional).
<code>origin</code>	Indicates the origin with respect to which the displacement and direction of the position are measured (optional).

12.5 Semantic relation classification schemes

12.5.1 Introduction

This subclause specifies classification schemes for describing semantic relations. Table 26 provides an overview of the semantic relation classification scheme. The semantic relations describe semantic information that relates entities such as the relationship of events in a narrative, or relationship of an object to multimedia content that depicts the object.

Table 33 — Overview of the classification schemes for semantic relations

<i>Tools</i>	<i>Functionality</i>
<code>SemanticRelation CS</code>	This classification scheme specifies semantic relations. The semantic relations describe semantic information that relates entities such as the relationship of events in a narrative, or relationship of an object to multimedia content that depicts the object.

12.5.2 SemanticRelation CS

12.5.2.1 Introduction

The `SemanticRelation CS` defines semantic relations. The semantic relations describe semantic information that relates entities such as the relationship of events in a narrative, or relationship of an object to multimedia content that depicts the object. In the following specification of the `SemanticRelation CS`, symbol *A* refers to the source of the relation and symbol *B* refers to the target of the relation.

12.5.2.2 SemanticRelation CS syntax

```
<!-- ##### -->
<!-- Definition of SemanticRelation CS (12.5.2) -->
<!-- ##### -->

<ClassificationScheme uri="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001">
  <Header xsi:type="DescriptionMetadataType">
    <Comment>
      <FreeTextAnnotation xml:lang="en">
```

```

        Semantic relations
    </FreeTextAnnotation>
</Comment>
</Header>
<Term termID="key">
    <Definition>
        A key B if and only if B is a key for accessing A.
    </Definition>
</Term>
<Term termID="keyFor">
    <Definition> Inverse relation of key </Definition>
</Term>
<Term termID="annotates">
    <Definition>
        A annotates B if and only if A is an annotation or description of B.
    </Definition>
</Term>
<Term termID="annotatedBy">
    <Definition> Inverse relation of annotates </Definition>
</Term>
<Term termID="shows">
    <Definition>
        A shows B if and only if A has an appearance in or is perceivable in B.
    </Definition>
</Term>
<Term termID="appearIn">
    <Definition> Inverse relation of shows </Definition>
</Term>
<Term termID="references">
    <Definition>
        A references B if and only if A refers to B and B does not appear in A.
    </Definition>
</Term>
<Term termID="referencedBy">
    <Definition> Inverse relation of references </Definition>
</Term>
<Term termID="quality">
    <Definition>
        A quality B if and only if B is a quality of one or more properties of A.
    </Definition>
</Term>
<Term termID="qualityOf">
    <Definition> Inverse relation of quality </Definition>
</Term>
<Term termID="symbolizes">
    <Definition>
        A symbolizes B if and only if A is a symbol of B.
    </Definition>
</Term>
<Term termID="symbolizedBy">
    <Definition> Inverse relation of symbolizes </Definition>
</Term>
<Term termID="location">
    <Definition>
        A location B if and only if B is a location of A.
    </Definition>
</Term>
<Term termID="locationOf">
    <Definition> Inverse relation of location </Definition>
</Term>

```

```

<Term termID="source">
  <Definition>
    A source B if and only if B is a source location of A.
  </Definition>
</Term>
<Term termID="sourceOf">
  <Definition> Inverse relation of source </Definition>
</Term>
<Term termID="destination">
  <Definition>
    A destination B if and only if B is a destination location of A.
  </Definition>
</Term>
<Term termID="destinationOf">
  <Definition> Inverse relation of destination </Definition>
</Term>
<Term termID="path">
  <Definition>
    A path B if and only if B is a path of A.
  </Definition>
</Term>
<Term termID="pathOf">
  <Definition> Inverse relation of path </Definition>
</Term>
<Term termID="time">
  <Definition>
    A time B if and only if B is a time of A.
  </Definition>
</Term>
<Term termID="timeOf">
  <Definition> Inverse relation of time </Definition>
</Term>
<Term termID="depicts">
  <Definition>
    A depicts B if and only if A is a depiction of B.
  </Definition>
</Term>
<Term termID="depictedBy">
  <Definition> Inverse relation of depicts </Definition>
</Term>
<Term termID="represents">
  <Definition>
    A represents B if and only if A is a representation of or symbolizes,
    in some sense, B.
  </Definition>
</Term>
<Term termID="representedBy">
  <Definition> Inverse relation of represents </Definition>
</Term>
<Term termID="context">
  <Definition>
    A context B if and only if B is a context or surrounding or
    contributing circumstances for A
  </Definition>
</Term>
<Term termID="contextFor">
  <Definition> Inverse relation of context </Definition>
</Term>
<Term termID="interprets">
  <Definition>

```


A interprets B if and only if A is an interpretation of B.
 </Definition>
 </Term>
 <Term termID="interpretedBy">
 <Definition> Inverse relation of interprets </Definition>
 </Term>
 <Term termID="agent">
 <Definition>
 A agent B if and only if B is an agent of or performs or initiates A.
 </Definition>
 </Term>
 <Term termID="agentOf">
 <Definition> Inverse relation of agent </Definition>
 </Term>
 <Term termID="patient">
 <Definition>
 A patient B if and only if B is affected by or undergoes the action of A.
 </Definition>
 </Term>
 <Term termID="patientOf">
 <Definition> Inverse relation of patient </Definition>
 </Term>
 <Term termID="experiencer">
 <Definition>
 A experiencer B if and only if B is an experiencer of or
 passively perceives a stimulus from A.
 </Definition>
 </Term>
 <Term termID="experiencerOf">
 <Definition> Inverse relation of experiencer </Definition>
 </Term>
 <Term termID="stimulus">
 <Definition>
 A stimulus B if and only if B is a stimulus of A.
 </Definition>
 </Term>
 <Term termID="stimulusOf">
 <Definition> Inverse relation of stimulus </Definition>
 </Term>
 <Term termID="causer">
 <Definition>
 A causer B if and only if B is a causer of A and B is not an agent of A
 </Definition>
 </Term>
 <Term termID="causerOf">
 <Definition> Inverse relation of causer </Definition>
 </Term>
 <Term termID="goal">
 <Definition>
 A goal B if and only if B is the goal or objective of A.
 </Definition>
 </Term>
 <Term termID="goalOf">
 <Definition> Inverse relation of goal </Definition>
 </Term>
 <Term termID="beneficiary">
 <Definition>
 A beneficiary B if and only if B is the beneficiary of A such as
 by some action of A.
 </Definition>

```

</Term>
<Term termID="beneficiaryOf">
  <Definition> Inverse relation of beneficiary </Definition>
</Term>
<Term termID="theme">
  <Definition>
    A theme B if and only if B is a theme or topic of A.
  </Definition>
</Term>
<Term termID="themeOf">
  <Definition> Inverse relation of theme </Definition>
</Term>
<Term termID="result">
  <Definition>
    A result B if and only if B is a result of A.
  </Definition>
</Term>
<Term termID="resultOf">
  <Definition> Inverse relation of result </Definition>
</Term>
<Term termID="instrument">
  <Definition>
    A instrument B if and only if B is an instrument of or is employed by A.
  </Definition>
</Term>
<Term termID="instrumentOf">
  <Definition> Inverse relation of instrument </Definition>
</Term>
<Term termID="accompanier">
  <Definition>
    A accompanier B if and only if B is an accompanier or joint agent of A,
    but B is not an initiator of A.
  </Definition>
</Term>
<Term termID="accompanierOf">
  <Definition> Inverse relation of accompanier </Definition>
</Term>
<Term termID="combination">
  <Definition>
    A0 combination A1, A2, ... An if and only if A0 is a combination of
    A1, A2, ... An.
  </Definition>
</Term>
<Term termID="summarizes">
  <Definition>
    A summarizes B if and only if A is a summary of B.
  </Definition>
</Term>
<Term termID="summarizedBy">
  <Definition> Inverse relation of summarizes </Definition>
</Term>
<Term termID="specializes">
  <Definition>
    A specializes B if and only if A is a specialization in meaning of B.
  </Definition>
</Term>
<Term termID="generalizes">
  <Definition> Inverse relation of specializes </Definition>
</Term>
<Term termID="similar">

```

```

    <Definition>
        A similar B if and only if A is similar to or synonymous with B.
    </Definition>
</Term>
<Term termID="opposite">
    <Definition>
        A opposite B if and only if A is opposite in meaning to B.
    </Definition>
</Term>
<Term termID="exemplifies">
    <Definition>
        A exemplifies B if and only if A is an example of B.
    </Definition>
</Term>
<Term termID="exemplifiedBy">
    <Definition> Inverse relation of exemplifies </Definition>
</Term>
<Term termID="interchangeable">
    <Definition>
        A interchangeable B if and only if A is interchangeable with or
        can be substituted for B.
    </Definition>
</Term>
<Term termID="identifier">
    <Definition>
        A identifier B if and only if A = B.
    </Definition>
</Term>
<Term termID="part">
    <Definition>
        A part B if and only if B is a part of A, or if and only if
        there exists a C such that (A union C) refines B. In other words,
        B is part of A if and only if A is a "meaningful" entity that is
        contained by B.
    </Definition>
</Term>
<Term termID="partOf">
    <Definition> Inverse relation of part </Definition>
</Term>
<Term termID="contrasts">
    <Definition>
        A contrasts B if and only if A is a contrast of B.
    </Definition>
</Term>
<Term termID="property">
    <Definition>
        A property B if and only if B is a property of if is used to qualify A.
    </Definition>
</Term>
<Term termID="propertyOf">
    <Definition> Inverse relation of property </Definition>
</Term>
<Term termID="user">
    <Definition>
        A user B if and only if B is a user of A.
    </Definition>
</Term>
<Term termID="userOf">
    <Definition> Inverse relation of user </Definition>
</Term>

```

```

<Term termID="component">
  <Definition>
    A component B if and only if B is a functional component of A.
  </Definition>
</Term>
<Term termID="componentOf">
  <Definition> Inverse relation of component </Definition>
</Term>
<Term termID="substance">
  <Definition>
    A substance B if and only if B is a substance or constituent of A,
    and B permeates A.
  </Definition>
</Term>
<Term termID="substanceOf">
  <Definition> Inverse relation of substance </Definition>
</Term>
<Term termID="entails">
  <Definition>
    A entails B if and only if B is a necessary condition for A to happen.
  </Definition>
</Term>
<Term termID="entailedBy">
  <Definition> Inverse relation of entails </Definition>
</Term>
<Term termID="manner">
  <Definition>
    A manner B if and only if B is a manner or adverb of A.
  </Definition>
</Term>
<Term termID="mannerOf">
  <Definition> Inverse relation of manner </Definition>
</Term>
<Term termID="state">
  <Definition>
    A state B if and only if B is a state of or provides the parameters
    to describe the state of A.
  </Definition>
</Term>
<Term termID="stateOf">
  <Definition> Inverse relation of state </Definition>
</Term>
<Term termID="influences">
  <Definition>
    A influences B if and only if B has a dependency on A, or in other words,
    there is a formula that relates the parameters of B to the parameters of A.
  </Definition>
</Term>
<Term termID="dependsOn">
  <Definition> Inverse relation of influences </Definition>
</Term>
<Term termID="membershipFunction">
  <Definition>
    A membershipFunction B if and only if A is a relation, and B
    is a membership function that governs the strength of A.
  </Definition>
</Term>
</ClassificationScheme>

```

12.5.2.3 SemanticRelation CS semantics

Table 34 defines the semantics of the relations in the SemanticRelation CS. Each entry in Table 34 identifies the name of the relation and its inverse relation, defines the relation mathematically, describes its properties (optional), and gives examples of its use. The inverse relations are defined as follows: "*B Relation C*" is equivalent to "*C InverseRelation B*", where *B* is the source and *C* is the target of the *Relation*, and *C* is the source and *B* is the target of the *InverseRelation*. This definition applies also in the case of each *n*-ary relation that has an inverse relation, where *B*=(*b1*, ..., *bn*) is the *n*-tuple of sources and *C*=(*c1*, ..., *cm*) is the *m*-tuple of the targets of the *Relation*.

Table 34 — Semantics of relations in the SemanticRelation CS

<i>Relation Name</i>	<i>Inverse Relation</i>	<i>Definition</i>	<i>Examples (Informative)</i>
key	keyFor	A key B if and only if B is a key for accessing A.	Video segment A key image B, which means that the image is a key for accessing the video segment.
annotates	annotatedBy	A annotates B if and only if A is an annotation or description of B.	Audio segment B annotates image A, which means that the audio segment provides an annotation or description of the image.
shows	appearsIn	A shows B if and only if A has an appearance in or is perceivable in B.	Image B shows object "car", which means that the car appears in the image.
references	referencedBy	A references B if and only if A refers to B and B does not appear in A.	Video B references object "car's engine", which means that the video refers to the car's engine.
quality	qualityOf	A quality B if and only if B is a quality of one or more properties of A.	Object "fire engine" quality red color, which means that the color red is a quality of fire engines.
symbolizes	symbolizedBy	A symbolizes B if and only if A is a symbol of B.	An image showing zeros and ones symbolizes object "computers", which means that the image is a symbol of computers.
location	locationOf	A location B if and only if B is a location of A.	In the description, "Mary receives a flower in the street," the event "receive" location semantic place "street", which means that the street is the location of the receiving event.

<i>Relation Name</i>	<i>Inverse Relation</i>	<i>Definition</i>	<i>Examples (Informative)</i>
source	sourceOf	A source B if and only if B is a source location of A.	In the example "John moved the box from the floor to table," the event "move" source semantic place "floor", which means that the floor is the source location of the moving event.
destination	destinationOf	A destination B if and only if B is a destination location of A.	In the description, "Mary moved from California to Toronto," the event "move" destination semantic place "Toronto", which means that Toronto is the destination location of the event move.
path	pathOf	A path B if and only if B is a path of A.	In the description "The train travelled along the track," event "travel" path semantic place "track", which means that the track is the path of the event travel.
time	timeOf	A time B if and only if B is a time of A.	In the description "Mary was born on Christmas," event "born" time semantic time "Christmas", which means that Christmas is the time of the birth event.
depicts	depictedBy	A depicts B if and only if A is a depiction of B.	Object "painting" depicts narrative world "glass with water", which means that the painting is a depiction of the glass with water.
represents	representedBy	A represents B if and only if A is a representation of or symbolizes in some sense B.	Object "Statue of Liberty" represents concept "liberty", which means that the Statue of Liberty represents the concept of liberty.
context	contextFor	A context B if and only if B is a context or surrounding contributing circumstances for A	Event "Rosa Parks' bus ride" context concept "civil rights struggle," which means that the civil rights struggle is the context for Rosa Parks' bus ride.

Relation Name	Inverse Relation	Definition	Examples (Informative)
interprets	interpretedBy	A interprets B if and only if A is an interpretation of B.	Concept "law enforcement" interprets event "policeman arresting a suspected killer", which means that law enforcement is an interpretation of the arrest event.
agent	agentOf	A agent B if and only if B is an agent of or performs or initiates A.	In the description "Peter gives the book to Mary," the event "give" agent object "Peter", which means that Peter is the agent of the giving.
patient	patientOf	A patient B if and only if B is affected by or undergoes the action of A.	In the description "Peter gives the book to Mary," the event "give" patient object "book", which means that the book is the patient of the giving.
experiencer	experiencerOf	A experiencer B if and only if B is an experiencer of or passively perceives a stimulus from A.	In the description "John hates pizza," the event "hate" experiencer object "John", which means that John is the experiencer of the hate.
stimulus	stimulusOf	A stimulus B if and only if B is a stimulus of A.	In the description, "The boy sees the barking dog," the event "see" stimulus object "dog", which means that the barking dog is the stimulus of the seeing.
causer	causerOf	A causer B if and only if B is a causer of A and B is not an agent of A	In the description "Lightening strikes the tree," the event "strike" causer object "lightening" means that lightening is the causer of the strike.
goal	goalOf	A goal B if and only if B is the goal or objective of A.	In the description "Mary receives a flower," the event "receive" goal object "Mary", which means that Mary is the goal of the receiving.
beneficiary	beneficiaryOf	A beneficiary B if and only if B is the beneficiary of A such as by some action of A.	In the description "John bakes a cake for Mary," the event "bake" beneficiary object "Mary", which means that Mary is the beneficiary of the baking.

<i>Relation Name</i>	<i>Inverse Relation</i>	<i>Definition</i>	<i>Examples (Informative)</i>
theme	themeOf	A theme B if and only if B is a theme or topic of A.	In the description "the boy talks about the dog," the event "talk" theme object "dog", which means that the dog is the theme of the boy's talking.
result	resultOf	A result B, if and only if B is a result of A.	In the description "John bakes a cake," the event "bake" result object "cake", which means that the cake is the result of the baking.
instrument	instrumentOf	A instrument B if and only if B is an instrument of or is employed by A.	In the description, "Mary cut the salami with a knife" the event "cut" instrument object "knife", which means that that the knife is the instrument of the cutting.
accompanier	accompanierOf	A accompanier B if and only if B is an accompanier or joint agent of A, but B is not an initiator of A.	In the description, "John travelled to France with Mary," the event "travel" accompanier object "Mary", which means that Mary is the accompanier of the travelling.
combination	-	A0 combination A1, A2, ... An if and only if A0 is a combination of A1, A2, ... A3.	Object "amphibious vehicle" combination objects "car," "boat," and "plane".
summarizes	summarizedBy	A summarizes B if and only if A is a summary of B.	In the description "Jim got hurt, he broke his leg and arm," the event "hurt" summarizes event "broke his leg and arm", which means that getting hurt is the summary of the breaking of a leg and an arm.
specializes	generalizes	A specializes B if and only if A is a specialization in meaning of B.	Object "dog" specializes object "mammal", which means that dog is a specialization of a mammal.
similar	similar	A similar B if and only if A is similar to or synonymous with B.	Object "car" similar object "automobile".

<i>Relation Name</i>	<i>Inverse Relation</i>	<i>Definition</i>	<i>Examples (Informative)</i>
opposite	opposite	A opposite B if and only if A is opposite in meaning to B.	Color "white" opposite color "black".
exemplifies	exemplifiedBy	A exemplifies B if and only if A is an example of B.	Object "Bill Clinton" exemplifies concept "former world leader", which means that Bill Clinton is an example of a former world leader.
interchangeable	interchangeable	A interchangeable B if and only if A is interchangeable with or can be substituted for B.	Object "car" interchangeable object "Cadillac," which means that the car and Cadillac are synonymous in the context of the description.
identifier	identifier	A identifier B if and only if A = B.	Agent object "Maria, 25 years old" identifier Agent object "Maria, 1 year old."
part	partOf	A part B if and only if B is a part of A, or in other words, if there exists a C such that (A union C) refines B. In other words, B is part of A if and only if A is a "meaningful" entity that is contained by B.	Semantic place "New York City" part semantic place "Manhattan," which means that Manhattan is part of New York City.
contrasts	contrasts	A contrasts B if and only if A is a contrast of B.	The color segments in "The Wizard of Oz" contrasts the sepia segments," which means that there is a contrast between the segments.
property	propertyOf	A property B if and only if B is a property of if is used to qualify A.	Object "banana" property concept "ripeness," which means that the banana is ripe.

Relation Name	Inverse Relation	Definition	Examples (Informative)
user	userOf	A user B if and only if B is a user of A.	The concept metaphor "blossoming" user concept "maturing".
component	componentOf	A component B if and only if B is a functional component of A.	Object "car" component object "wheel," which means that the wheel is a functional component of the car.
substance	substanceOf	A substance B if and only if B is a substance or constituent of A, and B permeates A.	object "martini" substance object "gin," which means that gin is a substance of a martini.
entails	entailedBy	A entails B if and only if B is a necessary condition for A to happen.	Event "divorce" entails event "marriage", which means that marriage is a necessary condition for divorce.
manner	mannerOf	A manner B if and only if B is a manner or adverb of A.	Event "talk" manner concept "whisper," which means that whispering is a manner of talking.
state	stateOf	A state B if and only if B is a state of or provides the parameters to describe the state of A.	In the description "The sunset is 90% orange," the semantic state event "sunset" state "90% orange".
influences	dependsOn	A influences B if and only if B has a dependency on A, or in other words, there is a formula that relates the parameters of B to the parameters of A.	Color B influences color A by the formula $A=0.2*B$, which means that color A has a dependency on color B.

Relation Name	Inverse Relation	Definition	Examples (Informative)
membershipFunction -		A memberhsipFunction B if and only if A is a relation, and B is a membership function that governs the strength of A.	The relation property between object "banana" and concept "ripeness" membershipFunction semantic state "cognitive color," which means that the cognitive color governs the strength of the ripeness of the banana.

12.5.2.4 SemanticRelation CS examples (informative)

The following example illustrates the use of the SemanticRelations CS and the Graph DS for describing the relation that object A agentOf event B.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Example 1 </Name>
      </Label>
      <SemanticBase xsi:type="ObjectType" id="objectA">
        <Label>
          <Name> Object A </Name>
        </Label>
      </SemanticBase>
      <SemanticBase xsi:type="EventType" id="eventB">
        <Label>
          <Name> Event B </Name>
        </Label>
      </SemanticBase>
      <Graph>
        <Node id="nodeA" href="#objectA"/>
        <Node id="nodeB" href="#eventB"/>
        <!-- Edge from a->b -->
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
          source="#nodeA" target="#nodeB"/>
      </Graph>
    </Semantics>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the SemanticRelation CS and the SemanticBase DS for describing the relation that object C causerOf event D.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Example 2 </Name>
      </Label>
      <SemanticBase xsi:type="ObjectType" id="objectC">
        <Label>
          <Name> Object C </Name>

```

```

    </Label>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:causerOf"
      target="#eventD"/>
  </SemanticBase>
  <SemanticBase xsi:type="EventType" id="eventD">
    <Label>
      <Name> Event D </Name>
    </Label>
  </SemanticBase>
</Semantics>
</Description>
</Mpeg7>

```

The following example illustrates the use of the Relation DS and several semantic relations for describing the semantics of an image showing a Mayan vessel as follows: "The vessel is an example of Maya art created in Guatemala in the 8th Century. The vessel's height is 14 cm and it has several paintings. The paintings show the realm of the lords of death with a death figure that dances and another figure that holds an axe and a handstone. The paintings represent sacrifice".

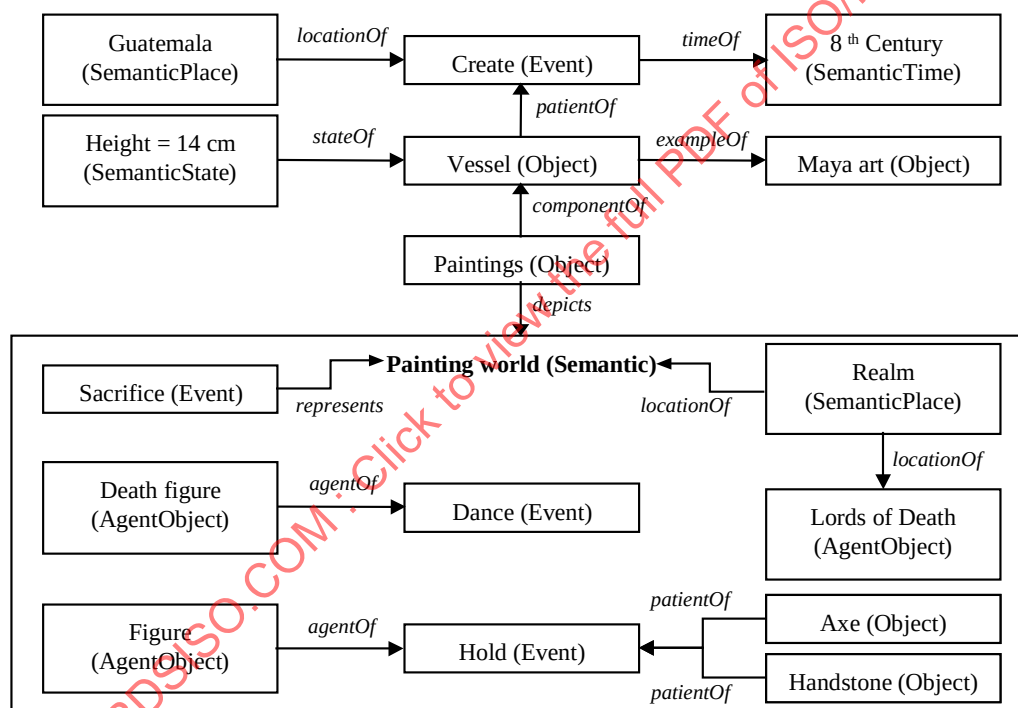


Figure 57 — Example semantic description of a Mayan vessel

The semantic description uses several semantic entities described using DSs derived from the SemanticBase DS as indicated within parenthesis in Figure 57 related by several semantic relations.

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name>Maya vessel</Name>
      </Label>
      <!-- Semantic entities -->
      <SemanticBase xsi:type="EventType" id="Create">
        <Label>
          <Name>Create</Name>

```

```

    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="SemanticPlaceType" id="Guatemala">
    <Label>
      <Name>Guatemala</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="SemanticTimeType" id="Century8">
    <Label>
      <Name>8th Century</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="ObjectType" id="Vessel">
    <Label>
      <Name>Vessel</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="ObjectType" id="MayaArt">
    <Label>
      <Name>Maya art</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="ObjectType" id="Paintings">
    <Label>
      <Name>Paintings</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="SemanticStateType" id="Height">
    <Label>
      <Name> Height </Name>
    </Label>
    <AttributeValuePair>
      <Attribute>
        <Name> Height </Name>
      </Attribute>
      <Unit>
        <Name> cm </Name>
      </Unit>
      <IntegerValue>14</IntegerValue>
    </AttributeValuePair>
  </SemanticBase>
  <SemanticBase xsi:type="SemanticType" id="PaintingWorld">
    <Label>
      <Name> World in the paintings </Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="SemanticPlaceType" id="Realm">
    <Label>
      <Name>Realm</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="AgentObjectType" id="DeathLords">
    <Label>
      <Name>Lords of Death</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="EventType" id="Sacrifice">
    <Label>
      <Name>Sacrifice</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="EventType" id="Dance">

```

```

    <Label>
      <Name>Dance</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="EventType" id="Hold">
    <Label>
      <Name>Hold</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="AgentObjectType" id="DeathFigure">
    <Label>
      <Name>Death figure</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="AgentObjectType" id="Figure">
    <Label>
      <Name>Figure</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="ObjectType" id="Axe">
    <Label>
      <Name>Axe</Name>
    </Label>
  </SemanticBase>
  <SemanticBase xsi:type="ObjectType" id="Handstone">
    <Label>
      <Name>Handstone</Name>
    </Label>
  </SemanticBase>
  <!-- Semantic relation graph within "PaintingWorld" -->
  <Graph>
    <Relation
      type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:locationOf"
      source="#Realm" target="#PaintingWorld"/>
    <Relation
      type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:representedBy"
      source="#Realm" target="#DeathLords"/>
    <Relation
      type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:represents"
      source="#Sacrifice" target="#PaintingWorld"/>
    <Relation
      type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
      source="#DeathFigure" target="#Dance"/>
    <Relation
      type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
      source="#Figure" target="#Hold"/>
    <Relation
      type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:patientOf"
      source="#Axe" target="#Hold"/>
    <Relation
      type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:patientOf"
      source="#Handstone" target="#Hold"/>
  </Graph>
</SemanticBase>
<!-- Semantic relation graph -->
<Graph>
  <Relation
    type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:locationOf"
    source="#Guatemala" target="#Create"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:timeOf"
    source="#Century8" target="#Create"/>

```

```

    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"
      source="#Vessel" target="#MayaArt"/>
    <Relation
      type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:componentOf"
      source="#Paintings" target="#Vessel"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:stateOf"
      source="#Height" target="#Vessel"/>
    <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:depicts"
      source="#Paintings" target="#PaintingWorld"/>
  </Graph>
</Semantics>
</Description>
</Mpeg7>

```

13 Navigation and access tools

13.1 Introduction

Clause 13 specifies description tools for navigation and access of multimedia content.

Table 35 — Overview of Navigation and Access Tools

<i>Tool</i>	<i>Functionality</i>
Summarization	Describes multimedia summaries and abstracts for efficient browsing and navigation of multimedia content (e.g. based on key video clips, key frames etc). See 13.2.
Partitions, Views and View Decompositions	Describes partitions, views and decompositions of image, video, and audio signals in space, time and frequency (e.g. low-resolution views or sub-regions of images). See 13.3.
Variations	Describes the relationships between different variations of multimedia content (e.g. conversion, revisions, summaries, etc). See 13.4.

13.2 Summarization

13.2.1 Introduction

Subclause 13.2 specifies tools for describing summaries that facilitate discovery, browsing, navigation, visualization and sonification of multimedia content. The Summarization tools enable fast and effective browsing and navigation of multimedia content by providing access to a set of multimedia summaries. Each multimedia summary is used to convey the essential information about the original (source) multimedia content. The following are the key summarization concepts.

- **Source content:** the original multimedia content that is being summarized, parts of which may be used in a summary. A source content item can be identified using one or more unique identifiers and can be located using one or more media locators (e.g., a URL and/or time stamps). One or more (distinct) source content items may be used in a single summary (e.g., a single audio summary can be created for multiple songs).
- **Summary content:** parts or components of the source content that are used to create a summary. Examples of such components are key video clips, key audio clips and key frames. These parts of the original content are located using time stamp elements, defined with respect to the time base of the source content. Summary content may also include alternative or external content that is not physically part of the source content to represent (parts of) the source content. Such content may be located using media locators (e.g., a URL).

- **Summary:** an abstract, extract or digest of the source content that conveys the essential information of the source content. A summary is a composite of the summary content that can be presented to a user. For example, a summary may simply be a concatenation of key video clips taken directly from the source content, where each clip is defined by its start time and duration. A summary may have its own time base, which allows for presenting the summary content at increased or decreased speeds. Multiple summaries can be specified for a single piece of multimedia content, without the need for generating or storing multiple versions of the source content.

Subclause 13.2 specifies the following tools.

Table 36 — Overview of Summarization Tools

<i>Tools</i>	<i>Functionality</i>
Hierarchical Summary	Describes summaries of time-varying audio, video or audiovisual data that support sequential and hierarchical navigation. Hierarchical summaries are composed of audio, video, or audiovisual clips, as well as their key frames and key sounds. For example, a highlight summary of television content can be composed using the HierarchicalSummary DS.
Sequential Summary	Describes summaries of time-varying audio, video or audiovisual data that support sequential navigation. Sequential summaries are composed of images or video-frames and audio clips, which can be synchronized and presented to the user at varying speeds. For example, a variable-speed fast-forward of video, or an audio-slideshow can be composed using the SequentialSummary DS.

An overview of the tools specified in subclause 13.2 is given in Figure 58. The Summarization DS is used to specify a set of summaries, where each summary is specified using the Summary DS. The HierarchicalSummary DS and the SequentialSummary DS each extend the Summary DS. The Summary DS is abstract.

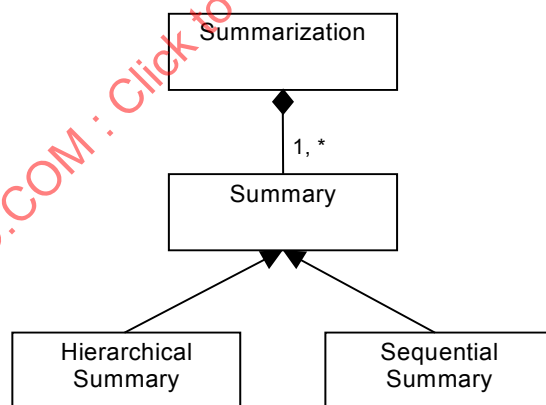


Figure 58 — Overview of Summarization tools

13.2.2 Summarization DS

13.2.2.1 Introduction

The Summarization DS is used to describe a set of summaries to enable rapid browsing, navigation, visualization and sonification of multimedia content. Each summary is specified using the Summary DS.

13.2.2.2 Summarization DS syntax

```

<!-- ##### -->
<!-- Definition of Summarization DS (13.2.2) -->
<!-- ##### -->

<!-- Definition of Summarization DS -->
<complexType name="SummarizationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Summary" type="mpeg7:SummaryType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

13.2.2.3 Summarization DS semantics

Semantics of the SummarizationType:

Name	Definition
SummarizationType	Specifies a set of Summary elements.
Summary	Describes a summary of multimedia content or a related group of summaries. See 13.2.3.

13.2.2.4 Summarization DS examples (informative)

The following example describes a set of two summaries: the first summary is a HierarchicalSummary, and the second summary is a SequentialSummary. Subelements of the HierarchicalSummary and SequentialSummary are introduced in later subclauses.

Note: XML comments are used in the examples below to replace description details.

```

<Summarization>
  <Summary xsi:type="HierarchicalSummaryType" components="keyVideoClips"
    hierarchy="independent">
    <SummarySegmentGroup>
      <SummarySegment> <!-- video clip #1 --> </SummarySegment>
      <SummarySegment> <!-- video clip #2 --> </SummarySegment>
    </SummarySegmentGroup>
  </Summary>
  <Summary xsi:type="SequentialSummaryType" components="audio">
    <AudioSummaryComponent> <!-- audio clip #1 --> </AudioSummaryComponent>
    <AudioSummaryComponent> <!-- audio clip #2 --> </AudioSummaryComponent>
  </Summary>
</Summarization>

```

13.2.3 Summary DS

13.2.3.1 Introduction

The Summary DS is used to describe a summary of multimedia content. The Summary DS is an abstract DS, from which the HierarchicalSummary DS and the SequentialSummary DS are derived. A summary may be associated with an entire multimedia content item or with a segment from the content, or with multiple content items. A summary may contain references to the description of the original (source) multimedia content being summarized, as well as an identifier and/or a locator of the original (source) multimedia content.

13.2.3.2 Summary DS syntax

```
<!-- ##### -->
<!-- Definition of Summary DS (13.2.3) -->
<!-- ##### -->

<!-- Definition of Summary DS -->
<complexType name="SummaryType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Name" type="mpeg7:TextualType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="SourceID" type="mpeg7:UniqueIDType"
          minOccurs="0"/>
        <element name="SourceLocator" type="mpeg7:TemporalSegmentLocatorType"
          minOccurs="0"/>
        <element name="SourceInformation" type="mpeg7:ReferenceType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

13.2.3.3 Summary DS semantics

Semantics of the SummaryType:

Name	Definition
SummaryType	An abstract DS from which the following description schemes are derived: • HierarchicalSummary DS. See 13.2.4 - 13.2.7. • SequentialSummary DS. See 13.2.8 - 13.2.11.
Name	Indicates an identifying or descriptive name of the summary. Multiple instances of this element may be present, where each element shall contain the same name expressed in different languages.
SourceID	Identifies the original (source) multimedia content that is being summarized. May be used if all the components in the summary are associated with source content with the same identifier.

Name	Definition
SourceLocator	Locates the original (source) multimedia content that is being summarized. May be used if all the components in the summary are associated with source content with the same locator.
SourceInformation	References an element of a description of the original (source) multimedia content.

13.2.4 HierarchicalSummary DS

13.2.4.1 Introduction

The HierarchicalSummary DS is used to describe summary content that can be grouped and organized into hierarchies to form multiple summaries. A diagram of the HierarchicalSummary DS is given in Figure 59. The HierarchicalSummary DS may be used to form a group of summaries, where each summary represents an alternative view of the source content and is represented using the SummarySegmentGroup DS. The SummarySegmentGroup DS is used to define a group of SummarySegments, which correspond to temporal segments of the source content. The HierarchicalSummary DS may be used to organize summaries into a tree-structured hierarchy, describing the content at multiple levels of detail, from coarse summaries to detailed summaries. Nodes in a hierarchy are described using the SummarySegmentGroup DS.

Note: a basic, yet functional, instance of the HierarchicalSummary DS may contain a single summary without any hierarchy.

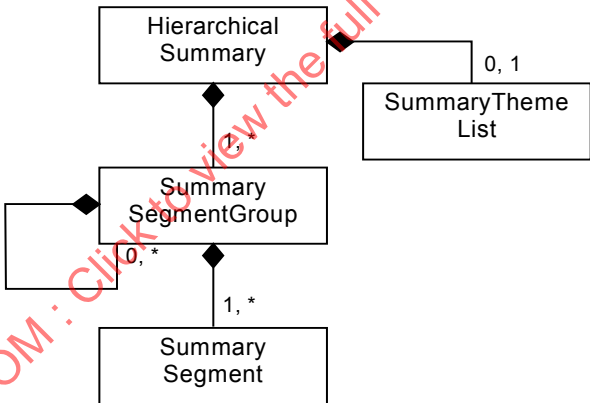


Figure 59 — The HierarchicalSummary DS

Summaries constructed using the HierarchicalSummary DS contain summary components, such as key AV clips, key video clips, key audio clips, key frames, key sounds and key themes. The types of components used in a HierarchicalSummary may be indicated using a list attribute. The components themselves are constructed using SummarySegmentGroup, SummarySegment and SummaryThemeList elements.

13.2.4.2 HierarchicalSummary DS syntax

```
<!-- ##### -->
<!-- Definition of HierarchicalSummary DS (13.2.4) -->
<!-- ##### -->

<!-- Definition of HierarchicalSummary DS -->
<complexType name="HierarchicalSummaryType">
  <complexContent>
    <extension base="mpeg7:SummaryType">
```

```

<sequence>
  <element name="SummaryThemeList" type="mpeg7:SummaryThemeListType"
    minOccurs="0"/>
  <choice minOccurs="1" maxOccurs="unbounded">
    <element name="SummarySegmentGroup"
      type="mpeg7:SummarySegmentGroupType"/>
    <element name="SummarySegmentGroupRef"
      type="mpeg7:ReferenceType"/>
  </choice>
</sequence>
<attribute name="components" use="optional">
  <simpleType>
    <list>
      <simpleType>
        <restriction base="NMTOKEN">
          <enumeration value="keyAudioVisualClips"/>
          <enumeration value="keyVideoClips"/>
          <enumeration value="keyAudioClips"/>
          <enumeration value="keyFrames"/>
          <enumeration value="keySounds"/>
          <enumeration value="keyThemes"/>
        </restriction>
      </simpleType>
    </list>
  </simpleType>
</attribute>
<attribute name="hierarchy" use="required">
  <simpleType>
    <restriction base="NMTOKEN">
      <enumeration value="independent"/>
      <enumeration value="dependent"/>
    </restriction>
  </simpleType>
</attribute>
</extension>
</complexContent>
</complexType>

```

13.2.4.3 HierarchicalSummary DS semantics

Semantics of the HierarchicalSummaryType:

Name	Definition
HierarchicalSummaryType	Specifies a group of summaries that contain summary components structured in a hierarchy. A HierarchicalSummary element contains SummarySegmentGroup elements, each of which defines a single, complete summary.
SummaryThemeList	Describes a list of textual themes associated with parts of the HierarchicalSummary. See 13.2.5.
SummarySegmentGroup	Describes a single summary, which can consist of a hierarchy of components. Each SummarySegmentGroup element in a HierarchicalSummary represents an alternative view on the content and may form the root node of a tree-structured hierarchy. A SummarySegmentGroup element may contain child SummarySegmentGroup elements, each forming a node in the tree structure. See 13.2.6.

Name	Definition
SummarySegmentGroupRef	References a SummarySegmentGroup element that describes a single summary. See SummarySegmentGroup above.
components	A list of the types of summary components included in a HierarchicalSummary (optional). The types allowed are defined as follows. • <i>keyAudioVisualClips</i> - The summaries shall contain key audiovisual clips, possibly ordered hierarchically. Such AV clips form an audiovisual summary of a particular duration. A key audiovisual clip can be an AV segment from the source content, or from external media. • <i>keyVideoClips</i> - The summaries shall contain key video clips, possibly ordered hierarchically. Such video clips form a video summary of a particular duration. A key video clip can be a video segment from the source content, or from external media. • <i>keyAudioClips</i> - The summaries shall contain key audio clips, possibly ordered hierarchically. Such audio clips form an audio summary of a particular duration. A key audio clip can be an audio segment from the source content, or from external media. • <i>keyFrames</i> - The summaries shall contain key frames, possibly ordered hierarchically. A summary may contain more key frames at each subsequent level of its hierarchy, to provide different levels of detail. A key frame can be a specific frame from a video segment, or an image that is not in the source content, possibly a synthetic image (pre-composed from multiple images). • <i>keySounds</i> - The summaries shall contain key sounds, possibly ordered hierarchically. A summary may contain more key sounds at each subsequent level of its hierarchy, to provide different levels of detail. A key sound may correspond to keywords in speech, sound effects, emotional sounds, exploding sounds, specific instrument sounds, and possibly synthetic sounds, either from the source content or from external media. • <i>keyThemes</i> - The summaries shall contain video clips, audio clips and/or AV clips, possibly ordered hierarchically, as well as textual descriptions of associated events or themes. Each summary is a collection of video clips and/or audio clips referring to particular key events or themes. Key events or themes may be described textually by keywords.
hierarchy	Indicates the type of the hierarchy with respect to the parent-child relationships between nodes at different levels of the tree-structured hierarchy. Nodes in the hierarchy are described using the SummarySegmentGroup DS. This attribute may be used to eliminate unnecessary duplication of information, i.e., SummarySegment elements, in the nodes of a tree. The types of a hierarchy are defined as follows. • <i>independent</i> - The information in the nodes on a single level of a hierarchy completely describes a particular summary, without reference to the information in the parent nodes of these nodes. Information in the parent nodes shall not be re-used in the children nodes. That is, the SummarySegment elements contained in SummarySegmentGroup elements at a single level of a tree shall be combined to form a summary at that level of detail.

Name	Definition
	• <i>dependent</i> - The information in children nodes in a hierarchy adds to, or refines, the information in the parent nodes. Information in the parent nodes shall be re-used in the children nodes. That is, the SummarySegment elements contained in all SummarySegmentGroup elements from the root to a particular level shall be combined to form a summary at that level of detail. Note: the value of this attribute may be ignored if none of the SummarySegmentGroup elements of a HierarchicalSummary contain children SummarySegmentGroup elements.

13.2.4.4 HierarchicalSummary DS examples (informative)

The HierarchicalSummary DS is built from temporal segments of audio, video or AV data, each of which is described by a SummarySegment. Each SummarySegment contains locators to the audio, video or AV data, to provide access to the associated key video clip or key audio clip, to key frames and to key sounds and may also contain textual annotation referring to key themes. SummarySegments are grouped into a single summary using the SummarySegmentGroup DS. For example, in Figure 60, the HierarchicalSummary contains two summaries, where the first summary consists of four SummarySegments and the second summary consists of three SummarySegments. Such summaries may correspond to two different themes and may provide alternative views of the source content, grouped together using the HierarchicalSummary DS. The description of this example is provided in subclause 13.2.6.

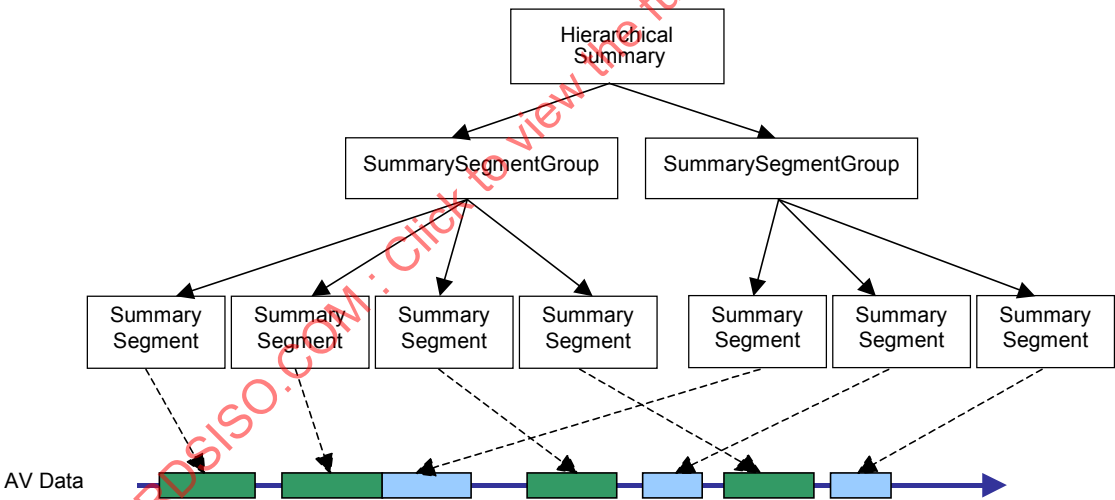


Figure 60 Illustration of a HierarchicalSummary containing two summaries, each described by a SummarySegmentGroup. In this example, each SummarySegment refers to a key audiovisual clip

The HierarchicalSummary DS can be used to summarize content at different levels of detail. The following is an example of a HierarchicalSummary description, which contains a summary with key video clips that are structured in multi-level fashion. The type of components in this summary is indicated using the components attribute, enabling applications to quickly identify the type of summary and the data contained therein. Since the value of the hierarchy attribute of this HierarchicalSummary is "dependent", elements of a hierarchical summary at level 1 add more video clips to the summary at level 0. Thus, each level in the hierarchy accumulates more information to provide a longer and more extensive video summary. The SourceLocator element refers to the original (source) video that is being summarized. The SourceInformation element refers to another description of the original (source) video. The actual summary components are specified using SummarySegment elements in the SummarySegmentGroup DS, specified in subclause 13.2.6.

```

<HierarchicalSummary components="keyVideoClips" hierarchy="dependent">
  <SourceLocator>      <!-- source video -->
    <MediaUri>file://disk/video001.mpg</MediaUri>
  </SourceLocator>
  <SourceInformation idref="videosegment001"/>
  <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>
</HierarchicalSummary>

```

13.2.5 SummaryThemeList DS

13.2.5.1 Introduction

The SummaryThemeList DS is used to describe a set of textual themes or events associated with summaries of multimedia content. Such themes may be associated with particular key frames, key video clips, key audio clips or with entire summaries containing sequences of such components. Each theme or event is labeled with an identifier, which can be referred to in other parts of a HierarchicalSummary description. Furthermore, one can represent a tree of themes by specifying a related, conceptually broader, theme as being the parent of a particular theme.

13.2.5.2 SummaryThemeList DS syntax

```

<!-- ##### -->
<!-- Definition of SummaryThemeList DS (13.2.5) -->
<!-- ##### -->

<!-- Definition of SummaryThemeList DS -->
<complexType name="SummaryThemeListType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="SummaryTheme" minOccurs="1" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="mpeg7:TextualType">
                <attribute name="id" type="ID"
                  use="required"/>
                <attribute name="parentID" type="IDREF"
                  use="optional"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

13.2.5.3 SummaryThemeList DS semantics

Semantics of the SummaryThemeListType:

Name	Definition
SummaryThemeListType	Defines a set of SummaryTheme elements.
SummaryTheme	Describes an event or theme in textual form, in terms of which a part of the content can be summarized.
id	Identifies the SummaryTheme element.
parentID	References another SummaryTheme element that corresponds to the parent or super-theme in a hierarchical index of themes (optional). Shall refer to the id attribute of a SummaryTheme element.

13.2.5.4 SummaryThemeList DS examples (informative)

The following is an example of a SummaryThemeList, which contains a set of themes (e.g., certain kinds of events) for a sports news program.

Note: this example describes a tree-structured representation of summary themes, as indicated by the parentID attributes. This structure is similar to an index and captures the themes in the multimedia content, allowing users to navigate the content accordingly. As shown in the last theme of the example ("John's favorite scenes"), a theme can be user-specific.

```
<SummaryThemeList>
  <SummaryTheme xml:lang="en" id="item0"> baseball </SummaryTheme>
  <SummaryTheme xml:lang="en" id="item01" parentID="item0">
    home run </SummaryTheme>
  <SummaryTheme xml:lang="en" id="item1"> basketball </SummaryTheme>
  <SummaryTheme xml:lang="en" id="item11" parentID="item1">
    three-pointer </SummaryTheme>
  <SummaryTheme xml:lang="en" id="item12" parentID="item1">
    slam dunk </SummaryTheme>
  <SummaryTheme xml:lang="en" id="item2"> soccer </SummaryTheme>
  <SummaryTheme xml:lang="en" id="item3"> John's favorite scenes </SummaryTheme>
</SummaryThemeList>
```

13.2.6 SummarySegmentGroup DS

13.2.6.1 Introduction

The SummarySegmentGroup DS is used to describe a summary (or parts of a summary) by referring to a sequence of segments from the source content (such as video, audio, or AV clips) with associated images (such as key frames) and sounds (such as key sounds). Each segment in a summary is described by a SummarySegment element. The SummarySegmentGroup DS can be used to construct video summaries, audio summaries, or audiovisual summaries. A SummarySegmentGroup may correspond to, for example, a summary with particular time duration or a particular set of events. A SummarySegmentGroup may contain children SummarySegmentGroup elements in order to construct summaries with more detail, i.e., summaries of longer duration, or summaries containing additional key frames. Thus, a coarse-to-fine hierarchy of summaries can be formed by embedding SummarySegmentGroup elements recursively, forming an ordered tree. Each child SummarySegmentGroup element is a node of the tree and has one parent; the root of the tree is a

SummarySegmentGroup element of the HierarchicalSummary. The SummarySegmentGroup DS contains a number of attributes to indicate the level of detail of the information contained within a SummarySegmentGroup element.

13.2.6.2 SummarySegmentGroup DS syntax

```
<!-- ##### -->
<!-- Definition of SummarySegmentGroup DS (13.2.6) -->
<!-- ##### -->

<!-- Definition of SummarySegmentGroup DS -->
<complexType name="SummarySegmentGroupType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Name" type="mpeg7:TextualType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="Caption" type="mpeg7:TextualType"
          minOccurs="0" maxOccurs="unbounded"/>
        <choice minOccurs="1" maxOccurs="unbounded">
          <element name="SummarySegment" type="mpeg7:SummarySegmentType"/>
          <element name="SummarySegmentRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SummarySegmentGroup"
            type="mpeg7:SummarySegmentGroupType"/>
          <element name="SummarySegmentGroupRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
      <attribute name="level" type="integer"
        use="optional"/>
      <attribute name="duration" type="mpeg7:mediaDurationType"
        use="optional"/>
      <attribute name="numOfKeyFrames" type="nonNegativeInteger"
        use="optional"/>
      <attribute name="fidelity" type="mpeg7:zeroToOneType"
        use="optional"/>
      <attribute name="themeIDs" type="IDREFS"
        use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

13.2.6.3 SummarySegmentGroup DS semantics

Semantics of the SummarySegmentGroupType:

Name	Definition
SummarySegmentGroupType	Specifies a single summary or part of a summary. A SummarySegmentGroup contains a set of SummarySegment elements that form (part of) a summary. A SummarySegmentGroup element may contain children SummarySegmentGroup elements, in which case a tree-structured hierarchy of summary components is formed. Each such tree has a single root node that is part of a HierarchicalSummary element and all nodes in a single tree describe the same summary (at different levels of detail).

Name	Definition
Name	Indicates an identifying or descriptive name of a summary or part of a summary specified by a SummarySegmentGroup element. Multiple instances of this element may be present, where each element shall contain the same name expressed in different languages.
Caption	Explanatory comment or caption accompanying a summary or part of a summary specified by a SummarySegmentGroup element. Multiple instances of this element may be present, where each element shall contain the same caption in different languages.
SummarySegment	Describes a summary segment by its key audiovisual clip, or key video clip and/or key audio clip, its key frames, key sounds and key themes. Multiple SummarySegment elements may be included in a SummarySegmentGroup element. The navigation or presentation order of SummarySegment elements within a summary is defined by the order of these elements in a SummarySegmentGroup element, unless explicitly indicated otherwise by order attributes in the SummarySegment elements. See 13.2.7.
SummarySegmentRef	References a SummarySegment element that describes a summary segment. See the definition of SummarySegment above.
SummarySegmentGroup	Describes a child SummarySegmentGroup that describes the current summary in more detail. Children SummarySegmentGroups are used to form a tree-structured hierarchy of summary components. A summary at a particular level of detail shall be constructed by combining the summary components from all children SummarySegmentGroup nodes at the same level in a single tree, and, optionally, adding the summary components from all their ancestor nodes. If the hierarchy of the hierarchical summary tree is "independent", the summary components (key audiovisual clips, key video clips, key audio clips, key frames and key sounds) in all SummarySegmentGroup nodes at the same level of a single tree shall be combined to define a single summary. If the hierarchy is "dependent", the summary components in all SummarySegmentGroup nodes at a particular level shall be added to the components of their ancestor nodes (in a single tree) to define a single, complete summary. In the latter case, all nodes up to a particular level in the tree contribute to a summary at a particular level of detail.
SummarySegmentGroupRef	References a SummarySegmentGroup element that describes the current summary in more detail. See SummarySegmentGroup above.
level	Indicates the level of a SummarySegmentGroup node in a hierarchy (optional). The root (SummarySegmentGroup) node in a hierarchy shall have level 0, its children (SummarySegmentGroup) nodes shall have level 1, and so forth.
duration	Indicates the total temporal duration of the key clips contained in the set of SummarySegment elements of a SummarySegmentGroup (optional). In case of an AV summary, shall indicate the total duration of key audiovisual clips. In case of a video summary, shall indicate the total duration of key video clips. In case of an audio summary, shall indicate the total duration of key audio clips.
numOfKeyFrames	Indicates the total number of key frames contained in the set of SummarySegment elements of a SummarySegmentGroup (optional).
fidelity	Indicates how well the information in a SummarySegmentGroup is represented by the information in its parent SummarySegmentGroup element, on a numerical scale between 0.0 and 1.0 (optional). Values closer to 1.0 correspond to better representations of this element by the associated parent element, while values closer to 0.0 correspond to worse representations.

Name	Definition
themeIDs	A list of references to SummaryThemes that correspond to key themes (key events) common to the SummarySegmentGroup element and all of its children SummarySegmentGroup and SummarySegment elements (optional). Each reference shall refer to the id attribute of a SummaryTheme element.

13.2.6.4 SummarySegmentGroup DS examples (informative)

The following is an example of a set of two video summaries referring to particular events in a program, in particular "slam dunks" and "three-point shots" in a basketball game. This description corresponds to the diagram shown in Figure 60. The first summary contains four AV clips, each showing a slam dunk; the second summary contains three AV clips, each showing a three-point shot. By grouping the clips into summaries of events, a user may choose to view only the clips of slam dunks; alternatively, the user may view all three-point shots.

Note: in this case, there is no notion of hierarchy in the underlying real-world events.

```
<HierarchicalSummary components="keyAudioVisualClips keyThemes"
  hierarchy="independent">
  <SourceInformation idref="mediainstance1"/>
  <SummaryThemeList>
    <SummaryTheme xml:lang="en" id="E0"> slam dunk </SummaryTheme>
    <SummaryTheme xml:lang="en" id="E1"> 3-point shot </SummaryTheme>
  </SummaryThemeList>
  <SummarySegmentGroup id="summary001" themeIDs="E0">
    <Name>Slam dunks</Name>
    <Caption>The L.A. Lakers scored four slam dunks in the second half.</Caption>
    <SummarySegment> <!-- segment #1 with slam dunk --> </SummarySegment>
    <SummarySegment> <!-- segment #2 with slam dunk --> </SummarySegment>
    <SummarySegment> <!-- segment #3 with slam dunk --> </SummarySegment>
    <SummarySegment> <!-- segment #4 with slam dunk --> </SummarySegment>
  </SummarySegmentGroup>
  <SummarySegmentGroup id="summary002" themeIDs="E1">
    <Name>3-Point shots</Name>
    <Caption>The L.A. Lakers scored three 3-point shots in this game.</Caption>
    <SummarySegment> <!-- segment #1 with 3-pt shot --> </SummarySegment>
    <SummarySegment> <!-- segment #2 with 3-pt shot --> </SummarySegment>
    <SummarySegment> <!-- segment #3 with 3-pt shot --> </SummarySegment>
  </SummarySegmentGroup>
</HierarchicalSummary>
```

The following is an example of a hierarchical video summary consisting of a set of video clips organized into two levels. At the top level (i.e., level 0), the summary has a duration of 40 seconds and consists of only two video clips. At the second level (i.e., level 1), the summary has a duration of 100 seconds and consists of five video clips.

Note: the coarse and the fine summary share two video clips, but these common clips are specified only once, by utilizing a dependent hierarchical structure (hierarchy is "dependent").

```
<HierarchicalSummary components="keyVideoClips" hierarchy="dependent">
  <SourceInformation idref="program001"/>
  <SummarySegmentGroup id="highlight001" level="0" duration="PT40S">
    <SummarySegment> <!-- segment #1 at level 0 --> </SummarySegment>
    <SummarySegment> <!-- segment #2 at level 0 --> </SummarySegment>
    <SummarySegmentGroup id="highlight002" level="1" duration="PT1M40S">
      <SummarySegment> <!-- segment #1 at level 1 --> </SummarySegment>
      <SummarySegment> <!-- segment #2 at level 1 --> </SummarySegment>
      <SummarySegment> <!-- segment #3 at level 1 --> </SummarySegment>
    </SummarySegmentGroup>
  </SummarySegmentGroup>
</HierarchicalSummary>
```

```

    </SummarySegmentGroup>
  </SummarySegmentGroup>
</HierarchicalSummary>

```

13.2.7 SummarySegment DS

13.2.7.1 Introduction

The SummarySegment DS is used to describe a temporal segment of multimedia content. As illustrated in Figure 61, the KeyAudioVisualClip element may be used to locate an AV component directly; alternatively, the KeyVideoClip element and KeyAudioClip element may be used to locate audio and visual components separately. Other alternatives are to use only the KeyVideoClip or only the KeyAudioClip element. The KeyAudioVisualClip element can be used in an AV summary. The KeyVideoClip can be used in an AV summary or in a video-only summary. The KeyAudioClip can be used in an AV summary or in an audio-only summary. If KeyVideoClip and KeyAudioClip are both used in a single SummarySegment, the audio and video data located should be presented to the user starting at the same time.

Furthermore, the KeyFrame and KeySound elements may be used to locate key frames and key sounds associated with the summary segment. Finally, key events or key themes associated with the segment can be defined using the themeIDs attribute.

The order attribute of the SummarySegment DS enables explicit specification of the navigation or presentation order of each SummarySegment DS in the description of a summary.

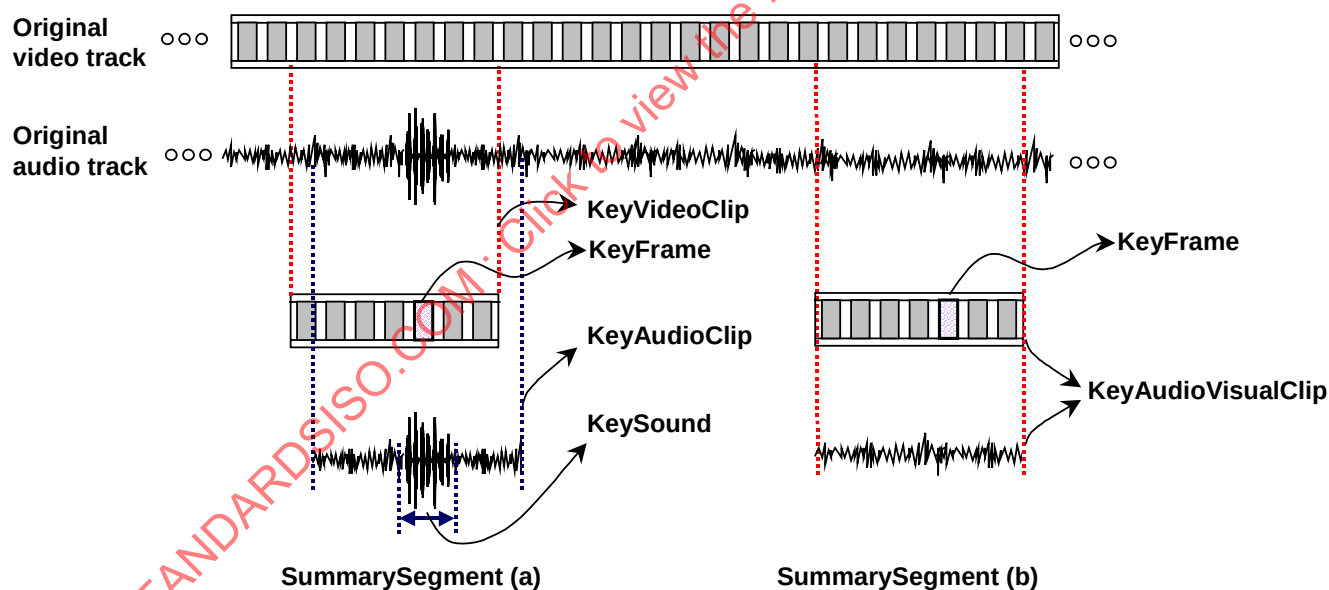


Figure 61 — Illustration of SummarySegment elements containing a KeyAudioVisualClip, KeyVideoClip, KeyAudioClip, KeyFrame and KeySound

13.2.7.2 SummarySegment DS syntax

```

<!-- ##### -->
<!-- Definition of SummarySegment DS (13.2.7) -->
<!-- ##### -->

```

```

<!-- Definition of SummarySegment DS -->
<complexType name="SummarySegmentType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Name" type="mpeg7:TextualType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="SourceID" type="mpeg7:UniqueIDType"
          minOccurs="0"/>
        <choice minOccurs="0">
          <element name="KeyAudioVisualClip"
            type="mpeg7:TemporalSegmentLocatorType"/>
          <sequence>
            <element name="KeyVideoClip"
              type="mpeg7:TemporalSegmentLocatorType" minOccurs="0"/>
            <element name="KeyAudioClip"
              type="mpeg7:TemporalSegmentLocatorType" minOccurs="0"/>
          </sequence>
        </choice>
        <element name="KeyFrame" type="mpeg7:ImageLocatorType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="KeySound" type="mpeg7:TemporalSegmentLocatorType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="themeIDs" type="IDREFS" use="optional"/>
      <attribute name="order" type="positiveInteger" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

13.2.7.3 SummarySegment DS semantics

Semantics of the SummarySegmentType:

Name	Definition
SummarySegmentType	Specifies a temporal segment of the content that is part of a summary. May contain an AV component (key audiovisual clip), a video component (key video clip), an audio component (key audio clip), images (key frame) or sounds (key sound).
Name	Indicates an identifying or descriptive name of a SummarySegment. Multiple instances of this element may be present, where each element shall contain the same name expressed in different languages.
SourceID	Identifies the original (source) content containing the segment that is being summarized. May be used if a single summary describes segments from multiple pieces of source content. If this element is absent, the SourceID element in HierarchicalSummary identifies the source content.
KeyAudioVisualClip	Locates a key audiovisual clip. A key audiovisual clip is an AV segment of the content, which can be used for navigation, browsing and summarization. See clause 6 for the definition of TemporalSegmentLocatorType. A KeyAudioVisualClip element may contain a MediaUri element. The MediaUri element may be used to locate AV data in case multiple pieces of AV content are being summarized in a single summary, or if external AV data (not related to the source AV content) is included in the summary. If the MediaUri element is absent, the SourceLocator element in HierarchicalSummary may be used to locate the source content instead. A KeyAudioVisualClip element may contain a MediaTime element. The

Name	Definition
	MediaTime element shall contain an element to define the start time of the AV clip in the source content, and may contain an element to define the duration of the AV clip. If the duration element is absent, the AV clip is defined to be open-ended, i.e., the AV clip lasts until the AV content ends.
KeyVideoClip	Locates a key video clip. A key video clip is a video segment of the content, which can be used for navigation, browsing and summarization. See clause 6 for the definition of TemporalSegmentLocatorType. For the semantics of the MediaUri and MediaTime elements included in this element, see KeyAudioVisualClip above.
KeyAudioClip	Locates a key audio clip. A key audio clip is an audio segment of the content, which can be used for navigation, browsing and summarization. See clause 6 for the definition of TemporalSegmentLocatorType. For the semantics of the MediaUri and MediaTime elements included in this element, see KeyAudioVisualClip above.
KeyFrame	Locates a key frame. A key frame is a single video frame in the content, which can be used for navigation, browsing and summarization. See clause 6 for the definition of ImageLocatorType. A KeyFrame element may contain a MediaUri element. For the semantics of the MediaUri element, see KeyAudioVisualClip above. A KeyFrame element may contain a media time element to define the time point of a single frame in the content.
KeySound	Locates a key sound. A key sound is a single sound in the content, which can be used for navigation, browsing and summarization. See clause 6 for the definition of TemporalSegmentLocatorType. A KeySound element may contain a MediaUri element. For the semantics of the MediaUri element, see KeyAudioVisualClip above. A KeySound element may contain a MediaTime element. The MediaTime element shall contain both a time point and a duration.
themeIDs	A list of references to SummaryThemes that correspond to key themes (key events) associated with the segment (optional). Each reference shall refer to the id attribute of a SummaryTheme element.
order	Indicates the navigation or presentation order of a SummarySegment element in a summary. SummarySegment elements with low order values shall be navigated or presented before elements with high order values. If this attribute is absent, the order of navigation or presentation is determined by the order of the SummarySegment element in a sequence of such elements. If present, the value of this attribute shall be unique among all SummarySegments forming a single summary.

13.2.7.4 SummarySegment DS examples (informative)

The HierarchicalSummary DS structures information in multimedia content into groups of video clips, audio clips, key frames and key sounds, possibly using a hierarchical set of levels. The resulting description allows navigating content by "zooming in" and "zooming out" to various levels of detail.

Figure 62 shows an example of a hierarchical key frame summary containing three levels. The key frame in each node is identified by A, B, and so forth, while fidelity values are denoted by e1, e2, and so forth. For example, the fidelity value e1 indicates how well the key frames in the subtree with root at node B is represented by the key frame at node A.

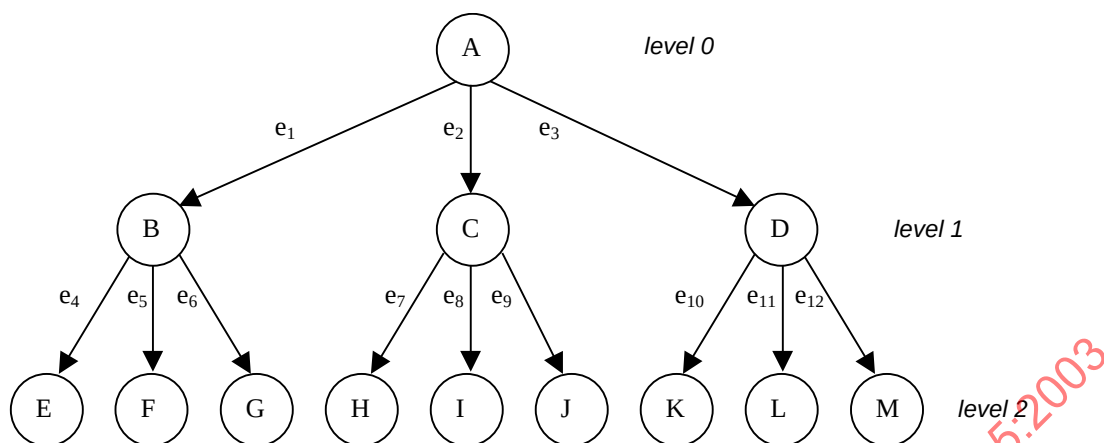


Figure 62 — An example of a key-frame hierarchy

The following is a description of the key-frame hierarchy in Figure 62 using the HierarchicalSummary DS. The fidelity values refer to Figure 62 and should be understood as example values. Note that a summary consisting of key frames at a particular level of detail shall be constructed by combining information from all SummarySegmentGroup nodes at the same depth in a single tree. In this example, the most detailed summary is constructed by combining the key frames from all leaf (level 2) SummarySegmentGroup nodes.

Note: children SummarySegmentGroup elements are specified after SummarySegment elements at each node in the hierarchy.

```

<HierarchicalSummary components="keyFrames" hierarchy="independent">
  <SourceInformation href="#segment001"/>
  <SummarySegmentGroup level="0" numOfKeyFrames="1">
    <SummarySegment id="key_frame_A">
      <KeyFrame><MediaTimePoint>T00:06:45</MediaTimePoint></KeyFrame>
    </SummarySegment>

    <SummarySegmentGroup level="1"
      numOfKeyFrames="1" fidelity="0.4"> <!-- e1 -->
      <SummarySegment id="key_frame_B">
        <KeyFrame><MediaTimePoint>T00:03:45</MediaTimePoint></KeyFrame>
      </SummarySegment>
      <SummarySegmentGroup level="2"
        numOfKeyFrames="1" fidelity="0.4"> <!-- e4 -->
        <SummarySegment id="key_frame_E">
          <KeyFrame><MediaTimePoint>T00:02:00</MediaTimePoint>
        </KeyFrame>
        </SummarySegment>
      </SummarySegmentGroup>
      <SummarySegmentGroup level="2"
        numOfKeyFrames="1" fidelity="0.2"> <!-- e5 -->
        <SummarySegment id="key_frame_F">
          <KeyFrame><MediaTimePoint>T00:03:00</MediaTimePoint>
        </KeyFrame>
        </SummarySegment>
      </SummarySegmentGroup>
      <SummarySegmentGroup level="2"
        numOfKeyFrames="1" fidelity="0.6"> <!-- e6 -->
        <SummarySegment id="key_frame_G">
          <KeyFrame><MediaTimePoint>T00:04:00</MediaTimePoint>
        </KeyFrame>
        </SummarySegment>
      </SummarySegmentGroup>
    </SummarySegmentGroup>
  </SummarySegmentGroup>

```



```

    </SummarySegmentGroup>
  </SummarySegmentGroup>

  <SummarySegmentGroup level="1"
    numOfKeyFrames="1" fidelity="0.5"> <!-- e2 -->
    <SummarySegment id="key_frame_C">
      <KeyFrame><MediaTimePoint>T00:07:15</MediaTimePoint></KeyFrame>
    </SummarySegment>
    <SummarySegmentGroup level="2"
      numOfKeyFrames="1" fidelity="0.3"> <!-- e7 -->
      <SummarySegment id="key_frame_H">
        <KeyFrame><MediaTimePoint>T00:06:00</MediaTimePoint>
      </KeyFrame>
    </SummarySegment>
  </SummarySegmentGroup>
  <SummarySegmentGroup level="2"
    numOfKeyFrames="1" fidelity="0.3"> <!-- e8 -->
    <SummarySegment id="key_frame_I">
      <KeyFrame><MediaTimePoint>T00:07:00</MediaTimePoint>
    </KeyFrame>
  </SummarySegment>
</SummarySegmentGroup>
<SummarySegmentGroup level="2"
  numOfKeyFrames="1" fidelity="0.8"> <!-- e9 -->
  <SummarySegment id="key_frame_J">
    <KeyFrame><MediaTimePoint>T00:08:00</MediaTimePoint>
  </KeyFrame>
</SummarySegment>
</SummarySegmentGroup>
</SummarySegmentGroup>

  <SummarySegmentGroup level="1"
    numOfKeyFrames="1" fidelity="0.2"> <!-- e3 -->
    <SummarySegment id="key_frame_D">
      <KeyFrame><MediaTimePoint>T00:10:40</MediaTimePoint></KeyFrame>
    </SummarySegment>
    <SummarySegmentGroup level="2"
      numOfKeyFrames="1" fidelity="0.3"> <!-- e10 -->
      <SummarySegment id="key_frame_K">
        <KeyFrame><MediaTimePoint>T00:10:00</MediaTimePoint>
      </KeyFrame>
    </SummarySegment>
  </SummarySegmentGroup>
  <SummarySegmentGroup level="2"
    numOfKeyFrames="1" fidelity="0.4"> <!-- e11 -->
    <SummarySegment id="key_frame_L">
      <KeyFrame><MediaTimePoint>T00:11:00</MediaTimePoint>
    </KeyFrame>
  </SummarySegment>
</SummarySegmentGroup>
<SummarySegmentGroup level="2"
  numOfKeyFrames="1" fidelity="0.1"> <!-- e12 -->
  <SummarySegment id="key_frame_M">
    <KeyFrame><MediaTimePoint>T00:12:00</MediaTimePoint>
  </KeyFrame>
</SummarySegment>
</SummarySegmentGroup>
</SummarySegmentGroup>
</SummarySegmentGroup>
</HierarchicalSummary>

```


The fidelity values of the SummarySegmentGroup elements in a hierarchical summary may be used by an application to adaptively select SummarySegmentGroup elements. For instance, a variable number of key frames can be extracted from the HierarchicalSummary in a scalable manner, given fidelity values for all elements with key frames in the hierarchy.

13.2.8 SequentialSummary DS

13.2.8.1 Introduction

The SequentialSummary DS is used to describe a single summary, which may contain a sequence of images or video frames, and may contain a sequence of audio clips, composing an audio, video or audiovisual slide-show or skim of the original (source) content. A sequential summary may also contain textual information associated with parts of the source content. A diagram of the SequentialSummary DS is given in Figure 63, which illustrates the three sequences (or tracks) of these elements. The VisualSummaryComponent D is used to describe the properties of a video frame, the AudioSummaryComponent D is used to describe the properties of an audio clip, and the TextualSummaryComponent D is used to describe associated textual information. The types of components used in a SequentialSummary are indicated using a list attribute.

The visual, audio and textual components may be synchronized to a common summary time line. The summary may refer to video frames and audio clips in the source content being summarized. Alternatively, the images or video frames and the audio clips that are part of the summary may be located separately from the source content. In particular, video frames may be located in a composite summary, i.e., a separate piece of existing content that represents a complete summary of the source content. Likewise, audio clips may be located in a composite summary.

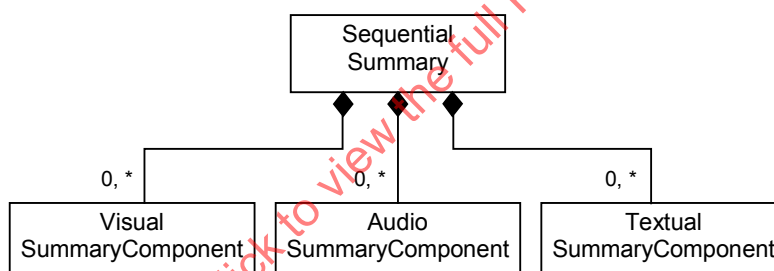


Figure 63 — The SequentialSummary DS

13.2.8.2 SequentialSummary DS syntax

```

<!-- ##### -->
<!-- Definition of SequentialSummary DS (13.2.8) -->
<!-- ##### -->

<!-- Definition of SequentialSummary DS -->
<complexType name="SequentialSummaryType">
  <complexContent>
    <extension base="mpeg7:SummaryType">
      <sequence>
        <choice>
          <element name="AudioVisualSummaryLocator"
            type="mpeg7:TemporalSegmentLocatorType" minOccurs="0"/>
          <sequence>
            <element name="VideoSummaryLocator"
              type="mpeg7:TemporalSegmentLocatorType" minOccurs="0"/>
            <element name="AudioSummaryLocator"
              type="mpeg7:TemporalSegmentLocatorType" minOccurs="0"/>
          
```

```

        </sequence>
      </choice>
      <element name="VisualSummaryComponent"
        type="mpeg7:VisualSummaryComponentType"
        minOccurs="0" maxOccurs="unbounded"/>
      <element name="AudioSummaryComponent"
        type="mpeg7:AudioSummaryComponentType"
        minOccurs="0" maxOccurs="unbounded"/>
      <element name="TextualSummaryComponent"
        type="mpeg7:TextualSummaryComponentType"
        minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="components" use="optional">
      <simpleType>
        <list>
          <simpleType>
            <restriction base="NMTOKEN">
              <enumeration value="visual"/>
              <enumeration value="audio"/>
              <enumeration value="textual"/>
            </restriction>
          </simpleType>
        </list>
      </simpleType>
    </attribute>
  </extension>
</complexContent>
</complexType>

```

13.2.8.3 SequentialSummary DS semantics

Semantics of the SequentialSummaryType:

Name	Definition
SequentialSummaryType	Specifies an audio, video or textual summary that may contain sequentially ordered images or video frames as well as audio clips and textual annotation. Each image or video frame and each audio clip may be described by certain properties relevant to the summary.
AudioVisualSummaryLocator	Locates an audiovisual summary, containing a composite of image or video frames and audio clips that represents a complete audiovisual summary of the source content. See clause 6 for the definition of TemporalSegmentLocatorType.
VideoSummaryLocator	Locates a video summary, containing a composite of images or video frames that represents a complete video summary of the source content. See clause 6 for the definition of TemporalSegmentLocatorType.
AudioSummaryLocator	Locates an audio summary, containing a composite of audio clips that represents a complete audio summary of the source content. See clause 6 for the definition of TemporalSegmentLocatorType.
VisualSummaryComponent	Describes temporal and spatial properties of an individual image or video frame of the original (source) content, which may also be part of the visual summary. See 13.2.9.

Name	Definition
AudioSummaryComponent	Describes temporal properties of an individual audio clip that is part of the original (source) content, which may also be part of the audio summary. See 13.2.10.
TextualSummaryComponent	Describes temporal properties of textual information associated with the original content and/or the summary. See 13.2.11.
components	A list of the types of summary components included in a SequentialSummary (optional). The types allowed are defined as follows. • <i>visual</i> - The summary shall contain visual components. A visual component may be an image or video frame. • <i>audio</i> - The summary shall contain audio components. An audio component may be an audio clip. • <i>textual</i> - The summary shall contain textual components. A textual component may be descriptive text, caption information, speech transcription or annotation related to the source content.

13.2.8.4 SequentialSummary DS examples (informative)

Using the SequentialSummary DS, a sequence of images, or frames from a video program can be shown sequentially in time - for example, as an animated slide show. The SequentialSummary DS also supports fast playback of parts of a video program, by referring to a separately stored composite summary of video frames. The following example of a video summary illustrates the SequentialSummary DS. The first locator specifies the location of the original (source) video, while the second locator specifies the location of a composite video summary. Then, each VisualSummaryComponent element describes the properties of a particular frame in the original video.

```
<SequentialSummary id="SoccerSummary001" components="visual">
  <SourceLocator>      <!-- source video -->
    <MediaUri>file:///disk/soccer/source/soccer001.mpg</MediaUri>
  </SourceLocator>
  <VideoSummaryLocator> <!-- summary video -->
    <MediaUri>file:///disk/soccer/summary/soccer001-summary.mpg</MediaUri>
  </VideoSummaryLocator>
  <VisualSummaryComponent> <!-- video frame #1 --> </VisualSummaryComponent>
  <VisualSummaryComponent> <!-- video frame #2 --> </VisualSummaryComponent>
  <VisualSummaryComponent> <!-- video frame #3 --> </VisualSummaryComponent>
  <VisualSummaryComponent> <!-- video frame #4 --> </VisualSummaryComponent>
</SequentialSummary>
```

Another example is illustrated in Figure 64, containing three tracks of elements associated with the multimedia content: image frames, textual annotation and audio clips. Each image frame is described by the VisualSummaryComponent D, each annotation is described by the TextualSummaryComponent D and each audio clip is described by the AudioSummaryComponent D.

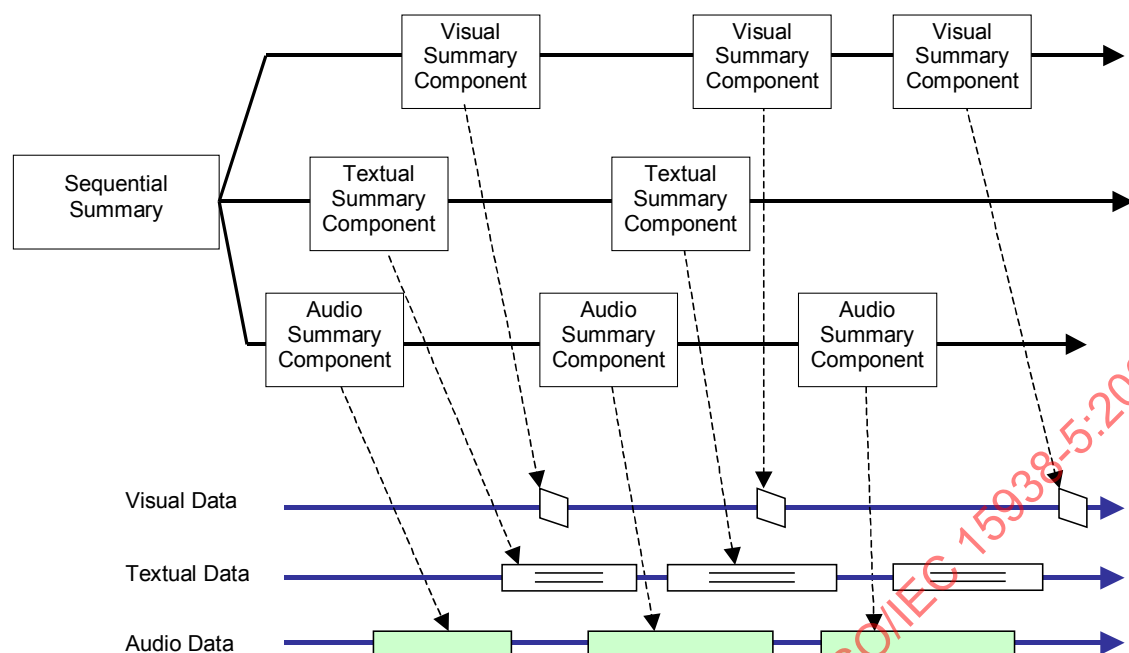


Figure 64 — Illustration of a Sequential Summary.

The following description corresponds to the diagram shown in Figure 64.

Note: the components attribute enables applications to quickly identify the type of summary and the data contained therein.

```
<SequentialSummary id="SoccerSummary005" components="visual audio textual">
  <SourceLocator>
    <!-- source video -->
    <MediaUri>file:///disk/soccer/source/soccer001.mpg</MediaUri>
  </SourceLocator>
  <VisualSummaryComponent> <!-- video frame #1 --> </VisualSummaryComponent>
  <VisualSummaryComponent> <!-- video frame #2 --> </VisualSummaryComponent>
  <VisualSummaryComponent> <!-- video frame #3 --> </VisualSummaryComponent>
  <AudioSummaryComponent> <!-- audio clip #1 --> </AudioSummaryComponent>
  <AudioSummaryComponent> <!-- audio clip #2 --> </AudioSummaryComponent>
  <AudioSummaryComponent> <!-- audio clip #3 --> </AudioSummaryComponent>
  <TextualSummaryComponent> <!-- text annotation #1 --> </TextualSummaryComponent>
  <TextualSummaryComponent> <!-- text annotation #2 --> </TextualSummaryComponent>
</SequentialSummary>
```

13.2.9 VisualSummaryComponent D

13.2.9.1 Introduction

The VisualSummaryComponent D is used to describe properties of an image or video frame summarizing the original (source) content. These properties may specify: the location of the image or video frame, possibly by referring to the source content; the relative amount of scene activity associated with the video frame; a region of interest in the image or video frame. The following diagram illustrates the role of several elements of this descriptor with respect to the source content and the summary.

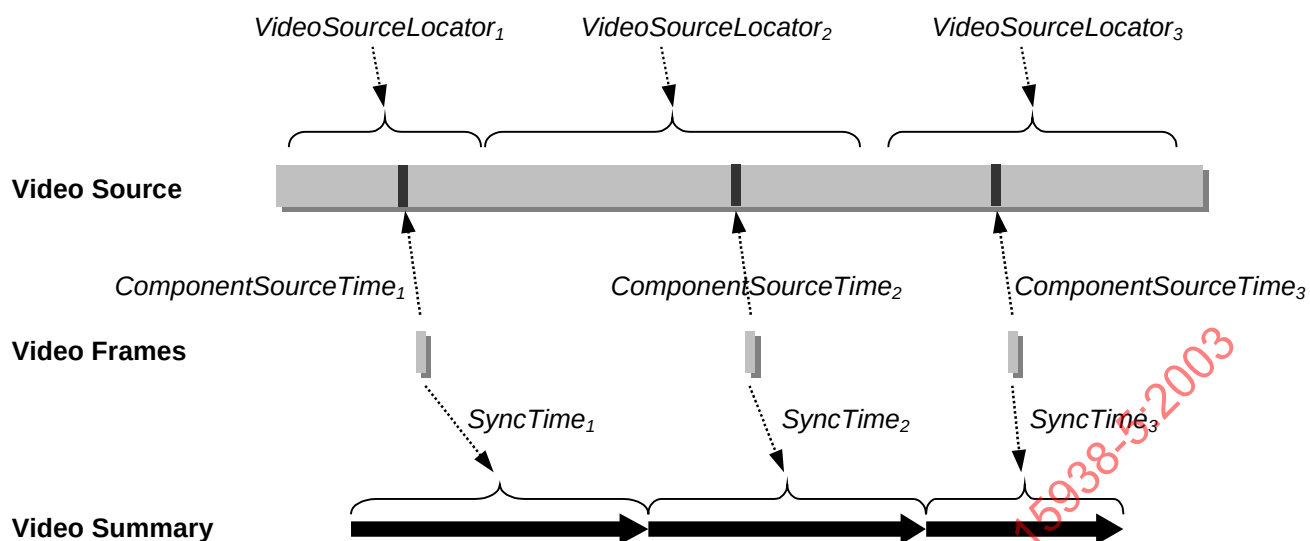


Figure 65 — Temporal properties of summary video frames and their relation to source and summary content

13.2.9.2 VisualSummaryComponent D syntax

```

<!-- ##### -->
<!-- Definition of VisualSummaryComponent D (13.2.9) -->
<!-- ##### -->

<!-- Definition of VisualSummaryComponent D -->
<complexType name="VisualSummaryComponentType">
  <complexContent>
    <extension base="mpeg7:DType">
      <sequence>
        <element name="VideoSourceID" type="mpeg7:UniqueIDType"
          minOccurs="0"/>
        <element name="VideoSourceLocator"
          type="mpeg7:TemporalSegmentLocatorType" minOccurs="0"/>
        <element name="ComponentSourceTime" type="mpeg7:MediaTimeType"
          minOccurs="0"/>
        <element name="ImageLocator" type="mpeg7:ImageLocatorType"
          minOccurs="0"/>
        <element name="SyncTime" type="mpeg7:MediaTimeType"
          minOccurs="0"/>
        <element name="FrameActivity" type="mpeg7:zeroToOneType"
          minOccurs="0"/>
        <element name="Region" type="mpeg7:RegionLocatorType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

13.2.9.3 VisualSummaryComponent D semantics

Semantics of the VisualSummaryComponentType:

Name	Definition
VisualSummaryComponentType	Describes temporal and spatial properties of an image or video frame summarizing the source content.
VideoSourceID	Identifies the source content containing the video segment that is being summarized. May be used if a single summary describes images or video frames from multiple pieces of source content. If this element is absent, the SourceID element in SequentialSummary identifies the source content.
VideoSourceLocator	Locates the original (source) video segment being summarized. May contain a MediaUri element with a URL locating the source content and/or a MediaTime element with the start time and duration of the video segment within the source content. The MediaUri element may be used if a single summary describes images or video frames from multiple pieces of source content. If the MediaUri element is absent, the SourceLocator element in SequentialSummary may be used to locate the source content instead. The MediaTime element may be used if different video segments of the same source content are being summarized in sequence. If neither the start time nor the duration is instantiated, the start time and duration of the video segment being summarized coincide with those of the source content. If the start time is instantiated but the duration is not, the end of the video segment being summarized coincides with the end of the source content.
ComponentSourceTime	Describes the time stamp of the video frame being referred to in the summary, within the source segment being summarized. May be used during presentation of a summary to switch from the summary to the source content.
ImageLocator	Locates an image or video frame that is part of the summary. May contain a MediaUri element with a URL to a separate file and/or a time stamp. If the MediaUri element is absent, the time stamp locates an image in the composite video summary of the parent SequentialSummary if present; otherwise, it locates an image in the source content.
SyncTime	Describes the start time and duration of the video frame in the summary. The presentation times of video frames in the summary, as specified by the SyncTime elements of different VisualSummaryComponent elements, shall not overlap in time. If the SyncTime element is not present, each component described by a VisualSummaryComponent element should be presented in the order that it appears in the description.
FrameActivity	Indicates the relative amount of scene activity between the previous video frame and the current frame in the original video content, on a numerical scale between 0.0 and 1.0. Values closer to 1.0 indicate high levels of activity in the scene, while values closer to 0.0 indicate low levels of activity. This information can be used for scalable key frame extraction, and smart fast forward.
Region	Locates a region of interest in a video frame from the original video content. If an ImageLocator element is used in combination with a Region element, the image referred to by the ImageLocator corresponds to this region of interest. The RegionLocatorType is defined in ISO/IEC 15938-3.

13.2.9.4 VisualSummaryComponent D examples (informative)

The following is an example of a SequentialSummary, enabling a smart quick view, which is a variable speed presentation of the video content that uses the scene activity within parts of the original content to control the speed of summary presentation. In this instance, I-frames from the source video are included in the summary. However, scene activity is computed for P-frames from the source video. Each VisualSummaryComponent element in this summary contains a ComponentSourceTime element that refers to a frame in the original video with a number expressed in units of 1/30th of a second. Some VisualSummaryComponent elements also refer to the corresponding frame in the composite video summary using an ImageLocator element. Finally, some frames have a FrameActivity value associated with it, indicating the relative amount of activity in the original video. In this example, a single piece of original (source) data is being summarized; therefore, a single SourceLocator element in the SequentialSummary is sufficient.

```
<SequentialSummary id="SoccerSummary006" components="visual">
  <SourceLocator>          <!-- Location of the source content -->
    <MediaUri>file://disk/soccer/source/soccer001.mpg</MediaUri>
  </SourceLocator>
  <VideoSummaryLocator>    <!-- Location of a composite video skim summary -->
    <MediaUri>file://disk/soccer/summary/soccer001-summary.mpg</MediaUri>
  </VideoSummaryLocator>

  <VisualSummaryComponent>  <!-- Properties of video I-frame-->
    <ComponentSourceTime>    <!-- Frame number in the original video -->
      <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
        mediaTimeBase="../../../../SourceLocator[1]">2
      </MediaRelIncrTimePoint>
    </ComponentSourceTime>
    <ImageLocator>          <!-- Locates summary frame (in video skim) -->
      <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
        mediaTimeBase="../../../../VideoSummaryLocator[1]">0
      </MediaRelIncrTimePoint>
    </ImageLocator>
  </VisualSummaryComponent>

  <VisualSummaryComponent>  <!-- Properties of video P-frame-->
    <ComponentSourceTime>    <!-- Frame number in the original video -->
      <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
        mediaTimeBase="../../../../SourceLocator[1]">5
      </MediaRelIncrTimePoint>
    </ComponentSourceTime>
    <FrameActivity>1.0</FrameActivity>    <!-- Frame activity value -->
  </VisualSummaryComponent>

  <VisualSummaryComponent>  <!-- Properties of video P-frame-->
    <ComponentSourceTime>    <!-- Frame number in the original video -->
      <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
        mediaTimeBase="../../../../SourceLocator[1]">8
      </MediaRelIncrTimePoint>
    </ComponentSourceTime>
    <FrameActivity>0.9</FrameActivity>    <!-- Frame activity value -->
  </VisualSummaryComponent>

  <VisualSummaryComponent>  <!-- Properties of video I-frame-->
    <ComponentSourceTime>    <!-- Frame number in the original video -->
      <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
        mediaTimeBase="../../../../SourceLocator[1]">11
      </MediaRelIncrTimePoint>
    </ComponentSourceTime>
    <ImageLocator>          <!-- Locates summary frame (in video skim) -->
      <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
```

```

        mediaTimeBase="../../VideoSummaryLocator[1]">1
    </MediaRelIncrTimePoint>
</ImageLocator>
</VisualSummaryComponent>

<VisualSummaryComponent>          <!-- Properties of video P-frame-->
    <ComponentSourceTime>          <!-- Frame number in the original video -->
        <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
            mediaTimeBase="../../SourceLocator[1]">14
        </MediaRelIncrTimePoint>
    </ComponentSourceTime>
    <FrameActivity>0.8</FrameActivity>      <!-- Frame activity value -->
</VisualSummaryComponent>

<VisualSummaryComponent>          <!-- Properties of video P-frame-->
    <ComponentSourceTime>          <!-- Frame number in the original video -->
        <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
            mediaTimeBase="../../SourceLocator[1]">17
        </MediaRelIncrTimePoint>
    </ComponentSourceTime>
    <FrameActivity>0.5</FrameActivity>      <!-- Frame activity value -->
</VisualSummaryComponent>

<VisualSummaryComponent>          <!-- Properties of video I-frame-->
    <ComponentSourceTime>          <!-- Frame number in the original video -->
        <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
            mediaTimeBase="../../SourceLocator[1]">20
        </MediaRelIncrTimePoint>
    </ComponentSourceTime>
    <ImageLocator>                <!-- Locates summary frame (in video skim) -->
        <MediaRelIncrTimePoint mediaTimeUnit="PT1N30F"
            mediaTimeBase="../../VideoSummaryLocator[1]">2
        </MediaRelIncrTimePoint>
    </ImageLocator>
</VisualSummaryComponent>
</SequentialSummary>

```

13.2.10 AudioSummaryComponent D

13.2.10.1 Introduction

The AudioSummaryComponent D is used to describe properties of an audio clip summarizing the original (source) content. These properties may specify the location of the audio clip, as a separate file, and/or possibly by referring to the source content.

13.2.10.2 AudioSummaryComponent D syntax

```

<!-- ##### -->
<!-- Definition of AudioSummaryComponent D (13.2.10) -->
<!-- ##### -->

<!-- Definition of AudioSummaryComponent D -->
<complexType name="AudioSummaryComponentType">
    <complexContent>
        <extension base="mpeg7:DType">
            <sequence>
                <element name="Title" type="mpeg7:TextualType"
                    minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```



```

        <element name="AudioSourceID" type="mpeg7:UniqueIDType"
            minOccurs="0"/>
        <element name="AudioSourceLocator"
            type="mpeg7:TemporalSegmentLocatorType" minOccurs="0"/>
        <element name="ComponentSourceTime" type="mpeg7:MediaTimeType"
            minOccurs="0"/>
        <element name="SoundLocator" type="mpeg7:TemporalSegmentLocatorType"
            minOccurs="0"/>
        <element name="SyncTime" type="mpeg7:MediaTimeType" minOccurs="0"/>
    </sequence>
</extension>
</complexContent>
</complexType>

```

13.2.10.3 AudioSummaryComponent D semantics

Semantics of the AudioSummaryComponentType:

Name	Definition
AudioSummaryComponent Type	Describes properties of an audio clip summarizing the source content.
Title	The textual title of the audio clip. This title describes the content of the audio clip. Typically it specifies the textual title of a song. Multiple instances of this element may be present, where each element shall contain the same title expressed in different languages.
AudioSourceID	Identifies the source content containing the audio segment that is being summarized. May be used if a single summary describes audio clips from multiple pieces of source content. If this element is absent, the SourceID element in SequentialSummary identifies the source content.
AudioSourceLocator	Locates the source audio segment being summarized. May contain a MediaUri element with a URL locating the source content and/or a MediaTime element with the start time and duration of the audio segment within the source content. The MediaUri element may be used if a single summary describes audio clips from multiple pieces of source content. If the MediaUri element is absent, the SourceLocator element in SequentialSummary may be used to locate the source content instead. The MediaTime element may be used if different audio segments of the same source content are being summarized in sequence. If neither the start time nor the duration is instantiated, the start time and duration of the audio segment being summarized coincide with those of the source content. If the start time is instantiated but the duration is not, the end of the audio segment being summarized coincides with the end of the source content.
ComponentSourceTime	Describes the start time and duration of the audio clip or audio slide being referred to in the summary, within the source segment being summarized. May be used during presentation of a summary to switch from the summary to the source content.
SoundLocator	Locates an audio clip that is part of the summary. May contain a MediaUri element with a URL to a separate file and/or a MediaTime element. If the MediaUri element is absent, the MediaTime element locates an audio clip in the composite audio summary of the parent SequentialSummary, if present; otherwise, it locates an audio clip in the source content.

Name	Definition
SyncTime	Describes the start time and duration of the audio clip in the summary. The presentation times of audio clips in the summary, as specified by the SyncTime elements of different AudioSummaryComponent elements, shall not overlap in time. If the SyncTime element is not present, each component described by a AudioSummaryComponent element should be presented in the order that it appears in the description.

13.2.10.4 AudioSummaryComponent D examples (informative)

The AudioSummaryComponent D can be used to provide an audio slide show (a succession of audio clips) and supports summarization of multiple pieces of content, such as a summary of multiple songs recorded on one CD album. In addition to playing audio slides (i.e., clips) one by one, applications can use the AudioSourceLocator element in order to switch playback between an audio slide component and its original. For example, one can listen to pieces of songs in the form of an audio slide show described by AudioSummaryComponent elements; and if one encounters some audio piece of his/her favorite song at a certain location in the song, one can listen to its original from that point forward.

There may be several types of audio summaries, depending on whether the original content is stored in a single stream of file, or in multiple streams or files. Also, each audio slide in the summary may be either: a) part of the original content, b) part of a composite summary, or c) in a separate stream or file. The following figure illustrates the case when there is a single source, but each audio summary component (audio clip) is located in a separate audio clip file.

In this case, the top-level SourceLocator associated with the entire summary indicates the location of a single source ("moz-req.aif"). AudioSourceLocator elements associated with each slide specify the location of audio clips (e.g. a song, a movement) using MediaTime elements (00m 00s to 05m 01s for the first part, and 05m 01s to 07m 41s for the second part).

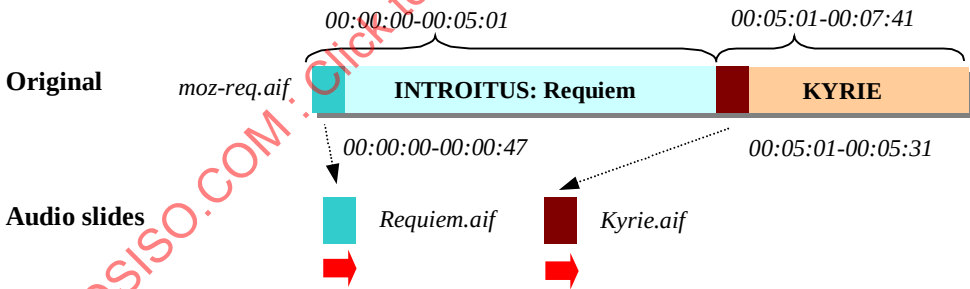


Figure 66 — Example of an audio summary with a single source

ComponentSourceTime elements indicate the start time and duration of audio summary components in the source, and SoundLocator elements each locate a separate audio clip using a MediaUri element ("Requiem.aif" and "Kyrie.aif" respectively).

```
<SequentialSummary id="classics004" components="audio">
  <Name>Mozart's Requiem KV 626</Name>          <!-- Summary name -->
  <SourceLocator>                                <!-- Location of the source content -->
    <MediaUri>file://Mozart/moz-req.aif</MediaUri>
  </SourceLocator>

  <AudioSummaryComponent>  <!-- Component #1 -->
    <Title xml:lang="de">INTROITUS: Requiem</Title>
    <AudioSourceLocator>    <!-- Location of movement #1 within the source -->
```

```

    <MediaTime>
      <MediaRelTimePoint
        mediaTimeBase="../../../../SourceLocator[1]">PT0S
      </MediaRelTimePoint>
      <MediaDuration>PT05M01S</MediaDuration>
    </MediaTime>
  </AudioSourceLocator>
  <ComponentSourceTime> <!-- Location of component #1 within the source -->
    <MediaRelTimePoint
      mediaTimeBase="../../../../SourceLocator[1]">PT0S
    </MediaRelTimePoint>
    <MediaDuration>PT47S</MediaDuration>
  </ComponentSourceTime>
  <SoundLocator> <!-- File location of slide component #1 -->
    <MediaUri>file://Mozart/Requiem.aif</MediaUri>
  </SoundLocator>
</AudioSummaryComponent>

<AudioSummaryComponent> <!-- Component #2 -->
  <Title xml:lang="de">KYRIE</Title>
  <AudioSourceLocator> <!-- Location of movement #2 within the source -->
    <MediaTime>
      <MediaRelTimePoint
        mediaTimeBase="../../../../SourceLocator[1]">PT05M01S
      </MediaRelTimePoint>
      <MediaDuration>PT02M40S</MediaDuration>
    </MediaTime>
  </AudioSourceLocator>
  <ComponentSourceTime> <!-- Location of component #2 within the source -->
    <MediaRelTimePoint
      mediaTimeBase="../../../../SourceLocator[1]">PT05M01S
    </MediaRelTimePoint>
    <MediaDuration>PT30S</MediaDuration>
  </ComponentSourceTime>
  <SoundLocator> <!-- File location of slide component #2 -->
    <MediaUri>file://Mozart/Kyrie.aif</MediaUri>
  </SoundLocator>
</AudioSummaryComponent>

<!-- More AudioSummaryComponent elements ... -->
</SequentialSummary>

```

The following figure illustrates the case when there are multiple sources, and the audio summary components (audio clips) are all part of a composite audio summary.

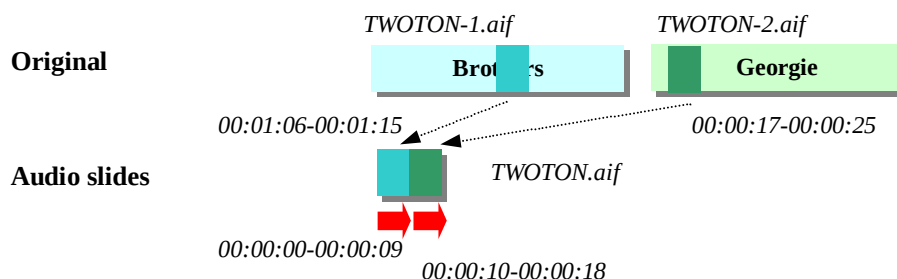


Figure 67 — Example of an audio summary with multiple sources

In this case, the AudioSummaryLocator element locates the composite audio summary ("TWOTON.aif"). The AudioSourceLocator elements in each AudioSummaryComponent element specify the location of each source using MediaUri ("TWOTON-1.aif" and "TWOTON-2.aif", respectively). ComponentSourceTime elements refer to the location of the audio clips within the source, and SoundLocator elements indicate the start time and duration of the audio clips within the composite audio summary using MediaTime (00s to 09s, and 10s to 18s, respectively). The Title element indicates the song title.

```
<SequentialSummary id="rock007" components="audio">
  <Name>Two Ton Shoe Rock Album</Name>          <!-- Summary name -->
  <AudioSummaryLocator>          <!-- Location of the composite audio summary -->
    <MediaUri>file://TwoTonShoe/Summary/TWOTON.aif</MediaUri>
  </AudioSummaryLocator>

  <AudioSummaryComponent>    <!-- Component #1 -->
    <Title xml:lang="en">Brothers</Title>
    <AudioSourceLocator>    <!-- Location of source #1 -->
      <MediaUri>file://TwoTonShoe/TWOTON-1.aif</MediaUri>
    </AudioSourceLocator>
    <ComponentSourceTime>    <!-- Location of component #1 within the source -->
      <MediaRelTimePoint
        mediaTimeBase="../../AudioSourceLocator[1]">PT01M06S
      </MediaRelTimePoint>
      <MediaDuration>PT09S</MediaDuration>
    </ComponentSourceTime>
    <SoundLocator>          <!-- Location of component #1 within audio summary -->
      <MediaTime>
        <MediaRelTimePoint
          mediaTimeBase="../../../AudioSummaryLocator[1]">PT0S
        </MediaRelTimePoint>
        <MediaDuration>PT09S</MediaDuration>
      </MediaTime>
    </SoundLocator>
  </AudioSummaryComponent>

  <AudioSummaryComponent>    <!-- Component #2 -->
    <Title xml:lang="en">Georgie</Title>
    <AudioSourceLocator>    <!-- Location of source #2 -->
      <MediaUri>file://TwoTonShoe/TWOTON-2.aif</MediaUri>
    </AudioSourceLocator>
    <ComponentSourceTime>    <!-- Location of component #2 within the source -->
      <MediaRelTimePoint
        mediaTimeBase="../../AudioSourceLocator[1]">PT17S
      </MediaRelTimePoint>
      <MediaDuration>PT08S</MediaDuration>
    </ComponentSourceTime>
    <SoundLocator>          <!-- Location of component #2 within audio summary -->
      <MediaTime>
        <MediaRelTimePoint
          mediaTimeBase="../../../AudioSummaryLocator[1]">PT10S
        </MediaRelTimePoint>
        <MediaDuration>PT08S</MediaDuration>
      </MediaTime>
    </SoundLocator>
  </AudioSummaryComponent>

  <!-- More AudioSummaryComponent elements ... -->
</SequentialSummary>
```

13.2.11 TextualSummaryComponent D

13.2.11.1 Introduction

The TextualSummaryComponent D is used to describe properties of textual information summarizing the original (source) content. For example, it can specify a textual table of contents of the source or any text that compactly describes the content. The textual information may, for example, be extracted from the captions or from the speech associated with the source content. These properties may specify: a textual string, the start and end time in the summary associated with such a string, and, possibly, references to a time interval in the original (source) content.

13.2.11.2 TextualSummaryComponent D syntax

```
<!-- ##### -->
<!-- Definition of TextualSummaryComponent D (13.2.11) -->
<!-- ##### -->

<!-- Definition of TextualSummaryComponent D -->
<complexType name="TextualSummaryComponentType">
  <complexContent>
    <extension base="mpeg7:DType">
      <sequence>
        <element name="AudioVisualSourceID" type="mpeg7:UniqueIDType"
          minOccurs="0"/>
        <element name="AudioVisualSourceLocator"
          type="mpeg7:TemporalSegmentLocatorType" minOccurs="0"/>
        <element name="ComponentSourceTime" type="mpeg7:MediaTimeType"
          minOccurs="0"/>
        <element name="FreeText" type="mpeg7:TextualType"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="SyncTime" type="mpeg7:MediaTimeType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

13.2.11.3 TextualSummaryComponent D semantics

Semantics of the TextualSummaryComponentType:

Name	Definition
TextualSummaryComponentType	Describes properties of textual information summarizing the source content.
AudioVisualSourceID	Identifies the source content containing the audio, video, or audiovisual segment that is being summarized. May be used if a single summary describes textual information associated with multiple pieces of source content. If this element is absent, the SourceID element in SequentialSummary identifies the source content.
AudioVisualSourceLocator	Locates the source segment being summarized. May contain a MediaUri element with a URL locating the source content and/or a MediaTime element with the start time and duration of the segment within the source content. The MediaUri element may be used if a single summary describes textual information associated with multiple pieces of source content. If the MediaUri element is absent, the SourceLocator element in SequentialSummary may be used to locate the source content instead. The MediaTime element

Name	Definition
	may be used if different segments of the same source content are being summarized in sequence. If neither the start time nor the duration is instantiated, the start time and duration of the segment being summarized coincide with those of the source content. If the start time is instantiated but the duration is not, the end of the segment being summarized coincides with the end of the source content.
ComponentSourceTime	Describes the start time and duration of the audio, video, or audiovisual data that is associated with the textual information described in the summary, within the source segment being summarized. May be used to specify the reference point for an element of a transcript, videotext, and so forth.
FreeText	A free text string containing textual information summarizing the source content. Multiple instances of this element may be present, where each element shall contain the same information expressed in different languages.
SyncTime	Describes the start time and duration of the temporal segment in the summary that is associated with the textual information. The SyncTime elements of different TextualSummaryComponent elements shall not overlap in time. If the SyncTime element is not present, each component described by a TextualSummaryComponent element should be presented in the order that it appears in the description.

13.2.11.4 TextualSummaryComponent D examples (informative)

The TextualSummaryComponent D can be used to synchronize text data with audio, video or audiovisual summaries. The following example shows the use of text to describe a table of contents of a presentation. The summary itself contains video frames, each associated with a particular video segment in the presentation, synchronized with a composite audio summary. Each video frame is stored in a separate (JPEG) file. For this example, start and end time of the video segments and their associated textual information in the summary video are as follows.

Topic	Start Time	End Time	Duration
"Introduction"	00:00:00	00:01:00	1 min
"Business Model"	00:01:01	00:03:00	2 min
"Stock Options"	00:03:01	00:05:00	2 min
"Get Rich Quick"	00:05:01	00:06:00	1 min

```
<SequentialSummary id="MSB4" components="visual audio textual">
  <AudioSummaryLocator>      <!-- Composite audio summary -->
    <MediaUri>http://www.mywebsite.com/CueVideo/audio_summary001.mp3</MediaUri>
    <MediaTime>
      <MediaTimePoint>T00:00:00</MediaTimePoint>
      <MediaDuration>PT6M</MediaDuration>
    </MediaTime>
  </AudioSummaryLocator>

  <VisualSummaryComponent>
    <ImageLocator>      <!-- Locates summary frame #1 -->
```

```

    <MediaUri>http://www.mywebsite.com/CueVideo/speaker1.jpg</MediaUri>
  </ImageLocator> <!-- Duration of frame #1 in summary video: 1 min -->
  <SyncTime>
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT01M</MediaDuration>
  </SyncTime>
</VisualSummaryComponent>
<VisualSummaryComponent>
  <ImageLocator> <!-- Locates summary frame #2 -->
    <MediaUri>http://www.mywebsite.com/CueVideo/speaker2.jpg</MediaUri>
  </ImageLocator> <!-- Duration of frame #2 in summary video: 4 min -->
  <SyncTime>
    <MediaTimePoint>T00:01:00</MediaTimePoint>
    <MediaDuration>PT04M</MediaDuration>
  </SyncTime>
</VisualSummaryComponent>
<VisualSummaryComponent>
  <ImageLocator> <!-- Locates summary frame #3 -->
    <MediaUri>http://www.mywebsite.com/CueVideo/speaker3.jpg</MediaUri>
  </ImageLocator> <!-- Duration of frame #3 in summary video: 1 min -->
  <SyncTime>
    <MediaTimePoint>T00:05:00</MediaTimePoint>
    <MediaDuration>PT01M</MediaDuration>
  </SyncTime>
</VisualSummaryComponent>

<TextualSummaryComponent>
  <FreeText xml:lang="en"> Introduction </FreeText>
  <SyncTime> <!-- Synchronizes text information with summary -->
    <MediaTimePoint>T00:00:00</MediaTimePoint>
    <MediaDuration>PT01M</MediaDuration>
  </SyncTime>
</TextualSummaryComponent>
<TextualSummaryComponent>
  <FreeText xml:lang="en"> Business Model </FreeText>
  <SyncTime> <!-- Synchronizes text information with summary -->
    <MediaTimePoint>T00:01:00</MediaTimePoint>
    <MediaDuration>PT02M</MediaDuration>
  </SyncTime>
</TextualSummaryComponent>
<TextualSummaryComponent>
  <FreeText xml:lang="en"> Stock Options </FreeText>
  <SyncTime> <!-- Synchronizes text information with summary -->
    <MediaTimePoint>T00:03:00</MediaTimePoint>
    <MediaDuration>PT02M</MediaDuration>
  </SyncTime>
</TextualSummaryComponent>
<TextualSummaryComponent>
  <FreeText xml:lang="en"> Get Rich Quick </FreeText>
  <SyncTime> <!-- Synchronizes text information with summary -->
    <MediaTimePoint>T00:05:00</MediaTimePoint>
    <MediaDuration>PT01M</MediaDuration>
  </SyncTime>
</TextualSummaryComponent>
</SequentialSummary>

```


13.3 Views, partitions and decompositions

13.3.1 Introduction

Subclause 13.3 specifies tools for describing views, partitions and decompositions of image, video, audio signals in space, time and frequency. The important concepts are defined as follows:

- **Partition:** corresponds to continuous multi-dimensional rectangular region defined in the space, time and/or frequency plane. A partition is used to provide the space and frequency coordinates of a View of an image, video or audio signal.
- **View:** corresponds to a region of an image, video or audio signal in multi-dimensional space, time and/or frequency, which is defined in terms of a space and frequency partition. For example, a low-resolution view of a segment of an audio signal provides a particular kind of View of the audio signal in time and temporal-frequency. The original image, video or audio signal is referred to as the source signal. The signal that corresponds to the view defined by the partition is referred to as the view signal.
- **View Decomposition:** corresponds to 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. For example, a spatial quad-tree decomposition of an image into a hierarchy of Views of the image provides a particular kind of tree-structured View Decomposition of the image in space.

In general, Views describe different partitions of image, video or audio signals in space and frequency, such as spatial regions of images, temporal-frequency subbands of audio, and 3-D wavelet subbands of video. View Decompositions describe different tree- and graph-structured decompositions of image, audio and video signals that provide different organizations of the Views. The View Decompositions describe both the node elements of the structured decompositions, which correspond to the Views, and the transition elements, which correspond to the analysis (decomposition) and synthesis (composition) dependencies among the Views.

The tools specified in subclause 13.3 include:

Table 37 — Overview of View, Partition and Decomposition Tools

<i>Tools</i>	<i>Functionality</i>
Partition	Describes space, time and/or frequency partitions that correspond to connected multi-dimensional rectangular regions in the space, time and/or frequency plane.
Filter	Describes filters and filtering operations applied to image, video and audio signals.
View	Describes partitioned views of image, video or audio signals.
ViewDecomposition	Describes sets of views or structured decompositions of views of image, video or audio signals.

Figure 68 illustrates the key concepts of Space and Frequency Views, Partitions and Decompositions. The basic entity of a View describes the view of an image, video or audio signal. The (original) source image, video or audio signal is referred to as the Source of the View, while the view signal is referred to as the Target of the View. While a Space and Frequency View is the most general case, other more specific types of Views are defined. The different types of Views may have properties that indicate the specific partitioning in the space and/or frequency plane, the filtering information for extracting the View from the frequency plane, and information for locating the view regions or segments in space or time relative to the Source signal. The different View Decompositions are formed as sets or as structured decompositions of such Views. The ViewSet describes a set of Views, which, for example, can be a complete and non-redundant set of subbands of an audio-signal, a set of wavelet subbands of an image, or an incomplete set of regions of an image. In addition, different structured decompositions based on trees or graphs can be described, using View trees and View graphs. Examples of structured decompositions include wavelet decompositions, tree-structured wavelet packet decompositions, spatial quad-trees, and space and frequency graphs.

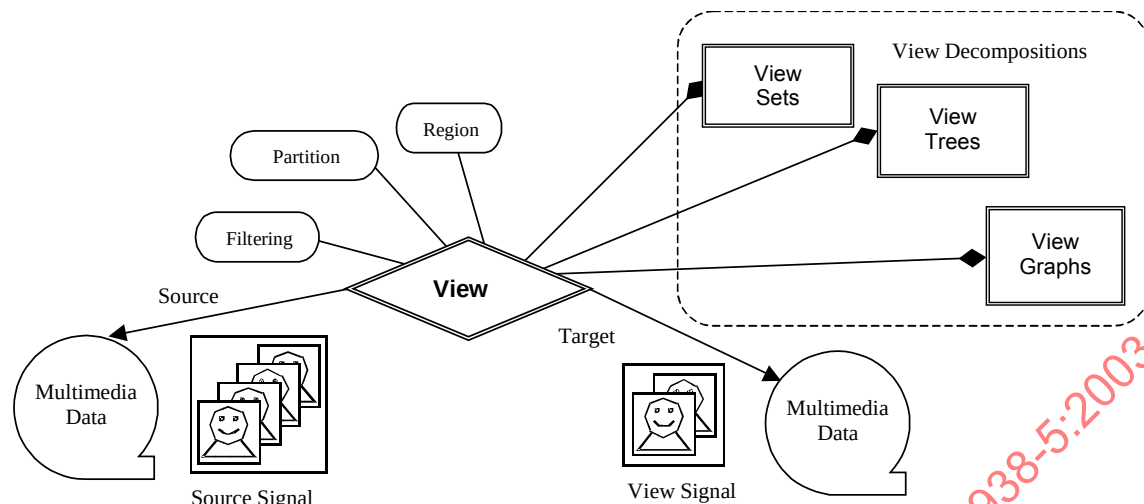


Figure 68 — Illustration of key concepts of Space and Frequency View, Partitions and Decompositions including Views, ViewSets and View Decompositions

13.3.2 Partition datatype

13.3.2.1 Introduction

Subclause 13.3.2 specifies tools for describing space and/or frequency partitions, which correspond to continuous multi-dimensional rectangular regions in the space, time and/or frequency plane. The Partition tool is used by the View tools to describe views of image, video and audio data in space and/or frequency.

The tools specified in subclause 13.3.2 include:

Table 38 — Overview of Partition Tools

Tools	Functionality
Partition	This tool describes a space and/or frequency partition that corresponds to a continuous multi-dimensional rectangular region in the space, time and/or frequency plane.
SignalPlane	This tool describes a point in the multi-dimensional space-time plane (abstract).
SignalFractionPlane	This tool describes a point in the multi-dimensional space-time plane using fractional units.
SignalSamplePlane	This tool describes a point in the multi-dimensional space-time plane using units of samples.
SignalPlaneOrigin	This tool describes the orientation of the axes in the multi-dimensional space by specifying the location of the origin.

The Partition concepts are illustrated in Figure 69, which shows different partitions of the 1-D time axis, 1-D temporal-frequency axis, and 2-D time-temporal-frequency plane. Figure 69.a illustrates a partition of the 1-D time axis covering the range in time from t_0 to t_1 , which is expressed in fractional units between zero and one. For example, the partition may be used to specify a temporal segment view of an audio signal. Figure 69.b illustrates a partition of the 1-D temporal-frequency axis covering the range in temporal-frequency from ω_0 to ω_1 , which is expressed in fractional units between zero and one. For example, the partition may be used to specify a subband view of an audio signal. Figure 69.c illustrates a rectangular partition of the 2-D time-temporal-frequency plane

covering the range in time from t_0 to t_1 and in temporal-frequency from ω_0 to ω_1 , which is expressed in fractional units between zero and one. For example, the partition may be used to specify a subband of a temporal segment of an audio signal. Similarly, a rectangular region in a 2-D image can be defined using a partition along the spatial axes (x and y) of the image.

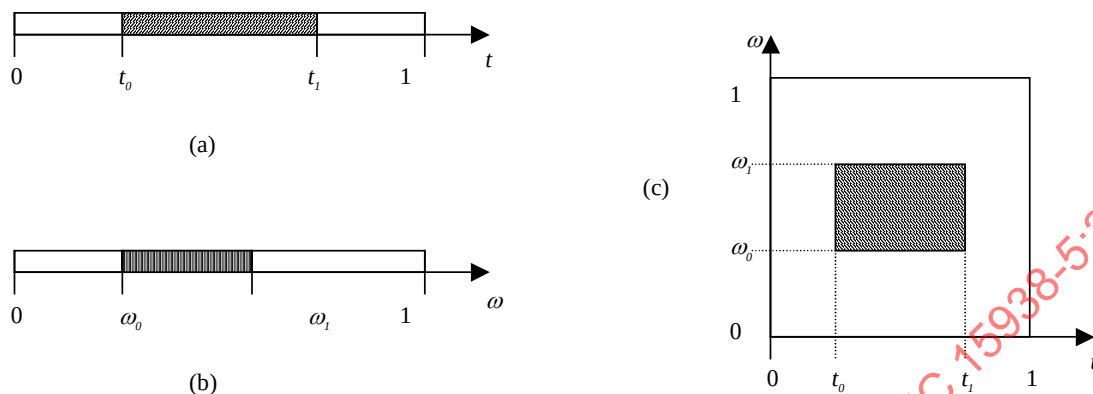


Figure 69 — Illustration of the different Partitions of time axis (a), temporal-frequency axis (b), and time / temporal-frequency plane (c)

The Partition datatype is used to describe a space and frequency partition that corresponds to continuous multi-dimensional rectangular region in the space and/or frequency plane. The Partition datatype provides a tool for describing the space and frequency coordinates of a view of an image, video and audio signal.

13.3.2.2 Partition datatype syntax

```
<!-- ##### -->
<!-- Definition of Partition Datatype (13.3.2) -->
<!-- ##### -->

<!-- Definition of Partition datatype -->
<complexType name="PartitionType">
  <sequence>
    <element name="Origin" type="mpeg7:SignalPlaneOriginType" minOccurs="0"/>
    <element name="Start" type="mpeg7:SignalPlaneType"/>
    <choice>
      <element name="End" type="mpeg7:SignalPlaneType"/>
      <element name="Extent" type="mpeg7:SignalPlaneType"/>
    </choice>
  </sequence>
  <attribute name="dim" use="optional">
    <simpleType>
      <restriction base="positiveInteger">
        <maxInclusive value="4"/>
      </restriction>
    </simpleType>
  </attribute>
</complexType>

<!-- Definition of SignalPlane datatype -->
<complexType name="SignalPlaneType" abstract="true">
  <sequence>
    <element name="Origin" type="mpeg7:SignalPlaneOriginType" minOccurs="0"/>
  </sequence>
  <attribute name="dim" use="optional">
    <simpleType>
```

```

        <restriction base="positiveInteger">
            <maxInclusive value="4"/>
        </restriction>
    </simpleType>
</attribute>
</complexType>

<!-- Definition of SignalPlaneFraction datatype -->
<complexType name="SignalPlaneFractionType">
    <complexContent>
        <extension base="mpeg7:SignalPlaneType">
            <attribute name="x" type="double" use="optional"/>
            <attribute name="y" type="double" use="optional"/>
            <attribute name="z" type="double" use="optional"/>
            <attribute name="t" type="double" use="optional"/>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of SignalPlaneSample datatype -->
<complexType name="SignalPlaneSampleType">
    <complexContent>
        <extension base="mpeg7:SignalPlaneType">
            <attribute name="x" type="integer" use="optional"/>
            <attribute name="y" type="integer" use="optional"/>
            <attribute name="z" type="integer" use="optional"/>
            <attribute name="t" type="integer" use="optional"/>
        </extension>
    </complexContent>
</complexType>

<!-- Definition of SignalPlaneOrigin datatype -->
<complexType name="SignalPlaneOriginType">
    <attribute name="xOrigin" use="optional" default="left">
        <simpleType>
            <restriction base="NMTOKEN">
                <enumeration value="left"/>
                <enumeration value="right"/>
            </restriction>
        </simpleType>
    </attribute>
    <attribute name="yOrigin" use="optional" default="bottom">
        <simpleType>
            <restriction base="NMTOKEN">
                <enumeration value="top"/>
                <enumeration value="bottom"/>
            </restriction>
        </simpleType>
    </attribute>
    <attribute name="zOrigin" use="optional" default="back">
        <simpleType>
            <restriction base="NMTOKEN">
                <enumeration value="back"/>
                <enumeration value="front"/>
            </restriction>
        </simpleType>
    </attribute>
    <attribute name="timeOrigin" use="optional" default="past">
        <simpleType>
            <restriction base="NMTOKEN">

```

```

        <enumeration value="past"/>
        <enumeration value="future"/>
    </restriction>
</simpleType>
</attribute>
</complexType>

```

13.3.2.3 Partition datatype semantics

Semantics of the PartitionType:

Name	Definition
PartitionType	Describes a multi-dimensional rectangle in space and time, which is used in the Space and Frequency Views to specify partitions in the space and/or frequency planes. A Partition is specified in the form of a multi-dimensional rectangle, which has starting and ending coordinates or starting coordinates and extent. The units for the coordinates shall be measured in terms of a fraction of the size of the space or in terms of the number of samples.
Origin	Describes the orientation of the axes in the multi-dimensional space by specifying the location of the origin (optional). If not specified, the default values of the SignalPlaneOriginType are assumed (see SignalPlaneOriginType below).
Start	Describes the multi-dimensional starting coordinate of the partition rectangle. The values of the elements of the Start coordinate are given in "units" along each dimension in the space or frequency plane.
End	Describes the multi-dimensional ending coordinate of the partition rectangle. The values of the elements of the End coordinate are given in "units" along each dimension in the space or frequency plane.
Extent	Describes the multi-dimensional size of the partition rectangle. The values of the elements of the Extent coordinate describe the size of the partition rectangle along each dimension in the space or frequency plane.
dim	Identifies the number of dimensions of the multi-dimensional rectangle that defines the Partition (optional). The value range of dim is from 1 to 4. For example, a rectangular partition of the 2-D spatial plane corresponding to an image has a dim of value 2.

Semantics of the SignalPlaneType:

Name	Definition
SignalPlaneType	Describes a point in the multi-dimensional space-time plane (abstract).
Origin	Describes the orientation of the axes in the multi-dimensional space by specifying the location of the origin (optional). If not specified, the default values of the SignalPlaneOriginType are assumed (see SignalPlaneOriginType below).
dim	Indicates the number of dimensions of the multi-dimensional space (optional). The value range of dim is from 1 to 4.

Semantics of SignalPlaneFractionType:

Name	Definition
SignalPlaneFractionType	Describes a point in the multi-dimensional space-time plane. SignalPlaneFractionType extends SignalPlaneType. Each element (x, y, z, t) is a coordinate that is expressed as a fractional value, relative to a finite plane defined by the Source signal. That is, the value 0.0 corresponds to the start of the space-, time-, or frequency plane, and 1.0 corresponds to the end of the plane.
x	Identifies a point in the x-axis of space (optional). For example, x can give the x-position or horizontal-position of a point in a 2-D image.
y	Identifies a point in the y-axis of space (optional). For example, y can give the y-position or vertical-position of a point in a 2-D image.
z	Identifies a point in the z-axis of space (optional). For example, z can give the z-position or depth of a point in a 3-D volume.
t	Identifies a point along a time axis (optional). For example, t can give the temporal position of a time-point in an audio signal.

Semantics of the SignalPlaneSampleType:

Name	Definition
SignalPlaneSampleType	Describes an integer point in the multi-dimensional space-time discrete lattice. SignalPlaneSampleType extends SignalPlaneType. Each element (x, y, z, t) is a coordinate that is measured in samples.
x	Identifies an integer point in the x-axis of discrete lattice (optional). For example, x can give the x-position or horizontal-position of a point in a 2-D image measured in samples.
y	Identifies an integer point in the y-axis of discrete lattice (optional). For example, y can give the y-position or vertical-position of a point in a 2-D image measured in samples.
z	Identifies an integer point in the z-axis of discrete lattice (optional). For example, z can give the z-position or depth of a point in a 3-D volume measured in samples.
t	Identifies an integer point along a time discrete lattice (optional). For example, t can give the temporal position of a time-point in an audio signal measured in samples.

Semantics of the SignalPlaneOriginType:

Name	Definition
SignalPlaneOriginType	Describes the orientation of the axes in the multi-dimensional space by specifying the location of the origin.
xOrigin	Identifies the location and orientation of the x-axis (optional). The values of this attribute are defined as follows: <i>left</i> – indicates that the origin of the x-axis is located on the left-side of the plane. This means the x-axis points from left to right. <i>right</i> – indicates that the origin of the x-axis is located on the right-side of the plane. This means the x-axis points from right to left. The default value of xOrigin is "left".
yOrigin	Identifies the location and orientation of the y-axis (optional). The values of this attribute are defined as follows: <i>top</i> – indicates that the origin of the y-axis is located on the upper-side of the plane. This means the y-axis points from top to bottom. <i>bottom</i> – indicates that the origin of the y-axis is located on the bottom-side of the plane. This means the y-axis points from bottom to top. The default value of yOrigin is "bottom".
zOrigin	Identifies the location and orientation of the z-axis (optional). The values of this attribute are defined as follows: <i>front</i> – indicates that the origin of the z-axis is located on the front-side of the plane. This means the z-axis points from front to back. <i>back</i> – indicates that the origin of the z-axis is located on the back-side of the plane. This means the z-axis points from back to front. The default value of zOrigin is "back".
timeOrigin	Identifies the location and orientation of the t-axis (optional). The values of this attribute are defined as follows: <i>past</i> – indicates that the origin of the t-axis is located on the past-side of the plane. This means the t-axis points from past to future. <i>future</i> – indicates that the origin of the t-axis is located on the future-side of the plane. This means the t-axis points from future to past. The default value of tOrigin is "past".

13.3.2.4 Partition datatype examples (informative)

The following examples illustrate descriptions of two-dimensional Partitions measured in samples and fractions. The first example specifies a two-dimensional partition, such as a partition of an image, which starts at sample or pixel (20, 40) and ends at sample (100, 100). This gives the Partition a size of (80 x 60) samples or pixels. The second example specifies a two-dimensional partition, which starts at position (0.0, 0.0) and has size (0.5, 0.5)

relative to the size of the space. For example, in describing a Partition of a (512 x 512) image, the Partition would start at pixel position (0, 0) and have size (256, 256) samples or pixels.

```
<Partition>
  <Origin xOrigin="left" yOrigin="top" />
  <Start xsi:type="SignalPlaneSampleType" x="20" y="40" />
  <End xsi:type="SignalPlaneSampleType" x="100" y="100" />
</Partition>

<Partition dim="2">
  <Origin xOrigin="left" yOrigin="bottom" />
  <Start xsi:type="SignalPlaneFractionType" x="0.0" y="0.0" />
  <Extent xsi:type="SignalPlaneFractionType" x="0.5" y="0.5" />
</Partition>
```

13.3.3 Filter datatype

13.3.3.1 Introduction

Subclauses 13.3.3 and 13.3.4 specify tools for describing filters and filtering operations applied to image, video and audio signals. Three types of filters are defined: 1-D filters, 2-D filters and multi-dimensional separable filters. The Filter datatype is an abstract type that forms the base of the specialized filter types. The Filter datatype is used to describe a discrete filter kernel, which when applied via convolution to an image, video or audio signal, attenuates the frequency content of the signal. The Filter1D datatype describes a 1-D filter. The Filter2D datatype describes a 2-D filter. The FilterSeparable datatype describes a multi-dimensional separable filter as a series of 1-D filters. The actual terms of the multi-dimensional separable filter are derived from the direct product of the 1-D filters.

The tools specified in subclauses 13.3.3 and 13.3.4 include:

<i>Tools</i>	<i>Functionality</i>
Filter	This tool describes a discrete filter (abstract).
Filter1D	This tool describes a discrete one-dimensional filter.
Filter2D	This tool describes a discrete two-dimensional filter.
FilterSeparable	This tool describes a discrete multi-dimensional separable filter.
Filtering	This tool describes the application of a filter operation to an image, video or audio signal.

The Filter concepts are illustrated in Figure 70, which shows the organization of the filter tools including FilterType, Filter1Dtype, Filter2Dtype, FilterSeparable, and Filtering.

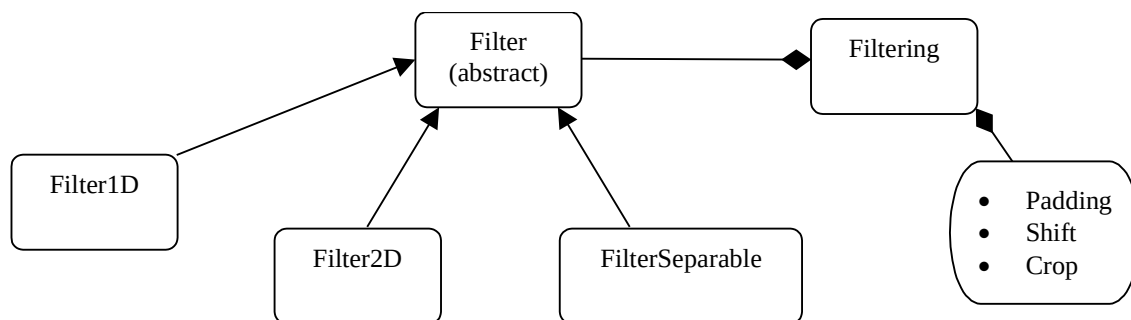


Figure 70 — Illustration of the Filter tools - FilterType is an abstract type from which Filter1DType, Filter2DType and FilterSeparableType are derived. FilteringType specifies the filtering of image, video and audio signals by specifying a Filter and the Padding, Shifting and Cropping parameters

13.3.3.2 Filter datatype syntax

```

<!-- ##### -->
<!-- Definition of Filter datatypes (13.3.3) -->
<!-- ##### -->

<!-- Definition of Filter datatype -->
<complexType name="FilterType" abstract="true">
  <attribute name="dim" type="positiveInteger" use="optional"/>
</complexType>

<!-- Definition of Filter1D datatype -->
<complexType name="Filter1DType">
  <complexContent>
    <extension base="mpeg7:FilterType">
      <sequence>
        <element name="Terms" type="mpeg7:DoubleMatrixType"/>
      </sequence>
      <attribute name="phase" type="integer" use="optional" default="0"/>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of Filter2D datatype -->
<complexType name="Filter2DType">
  <complexContent>
    <extension base="mpeg7:FilterType">
      <sequence>
        <element name="Terms" type="mpeg7:DoubleMatrixType"/>
      </sequence>
      <attribute name="xPhase" type="integer" use="optional" default="0"/>
      <attribute name="yPhase" type="integer" use="optional" default="0"/>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of FilterSeparable datatype -->
<complexType name="FilterSeparableType">
  <complexContent>
    <extension base="mpeg7:FilterType">
      <sequence>
        <element name="Filter1D" type="mpeg7:Filter1DType"
  
```



```

        minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
</extension>
</complexContent>
</complexType>

```

13.3.3.3 Filter datatype semantics

Semantics of the `FilterType`:

Name	Definition
<code>FilterType</code>	Describes a discrete filter (abstract).
<code>dim</code>	Identifies the dimensionality of the filter (optional). For example, <code>dim</code> has the value 1 for a 1-D filter and the value 2 for a 2-D filter.

Semantics of the `Filter1DType`:

Name	Definition
<code>Filter1DType</code>	Describes a discrete 1-D filter. If a value of <code>dim</code> is specified for <code>Filter1DType</code> , then it shall have a value of 1.
<code>Terms</code>	Describes the terms of the discrete 1-D filter. <code>Terms</code> is a 1-D matrix containing a vector of doubles. The length of the 1-D filter is given by the <code>mpeg7:dim</code> attribute of the <code>Terms</code> vector. The <code>Terms</code> vector gives the value of the discrete impulse response function of the discrete 1-D filter.
<code>phase</code>	Identifies the phase of the 1-D filter by specifying the position of the origin (optional). The origin is given relative to the first element in the <code>Terms</code> vector. By default, the phase is zero, which gives the origin at the first element.

Semantics of the `Filter2DType`:

Name	Definition
<code>Filter2DType</code>	Describes a discrete 2-D filter. If a value of <code>dim</code> is specified for <code>Filter2DType</code> , then it shall have a value of 2.
<code>Terms</code>	Describes the terms of the discrete 2-D filter. <code>Terms</code> is a 2-D matrix of doubles. The size of the 2-D filter is given by the <code>mpeg7:dim</code> attribute of the <code>Terms</code> matrix. The two dimensions of the matrix map to the dimensions of an image or video as follows: the rows of the matrix map to the rows of the image or video and the columns of the matrix map to the columns of the image or video. The 2-D filter shall not be applied to 1-D audio signals. The <code>Terms</code> matrix gives the value of the discrete impulse response array of the 2-D discrete filter.
<code>xPhase</code>	Identifies the phase of the 2-D filter along the x dimension by specifying the x-position of the origin (optional). The x-position of the origin is given relative to the first column in the <code>Terms</code> array. By default, the <code>xPhase</code> is zero, which gives the x-position of the origin at the first column.
<code>yPhase</code>	Identifies the phase of the 2-D filter along the y dimension by specifying the y-position of the origin (optional). The y-position of the origin is given relative to the first row in the <code>Terms</code> array. By default, the <code>yPhase</code> is zero, which gives the y-position of the origin at the first row.

Semantics of the FilterSeparableType:

Name	Definition
FilterSeparableType	Describes a discrete multi-dimensional separable filter. If a value of dim is specified for FilterSeparableType, then it shall have a value equivalent to the number of 1-D filters in the sequence of Filter1D elements.
Filter1D	Describes a series of 1-D filters that form a separable filter. The actual terms of the multi-dimensional separable filter shall be derived via the direct product of the 1-D filters. The 1-D filters shall be given in the following order: x, y, z, t, where z is optional, where x refers to the rows of an image or video, and y refers to the columns of an image or video, z refers to the depth dimension of 3-D data, and t refers to the time dimension in audio or video. For example, given a separable 3-D subband filter (three 1-D filters) applied to video, the individual filters in the sequence shall be given in the order x, y, t.

13.3.3.4 Filter datatype examples (informative)

The following example describes a 1-D discrete filter. The terms of the filter have values 0.5 and 0.5. The example defines a two-tap averaging filter, which when applied to audio takes the average of neighboring samples.

```
<Filter1D>
  <Terms mpeg7:dim="2"> 0.5 0.5 </Terms>
</Filter1D>
```

The following example describes a 2-D discrete filter. The filter corresponds to a 3 x 3 Laplacian filter, which when applied to an image takes the difference of pixel values vertically and horizontally in the neighborhood around each pixel. The Laplacian filter is commonly used for edge detection.

```
<Filter2D>
  <Terms mpeg7:dim="3 3">
    0.0 -1.0 0.0
    -1.0 4.0 -1.0
    0.0 -1.0 0.0
  </Terms>
</Filter2D>
```

The following example describes a discrete multi-dimensional separable filter. The individual filters that make up the separable filter are specified as two three-tap one-dimensional filters. The terms of the 2-D separable filter are derived from the direct product of the 1-D filters to produce the 2-D separable filter. The derived terms of the 2-D separable filter (top) are equivalent to those of the 2-D filter specified below it (bottom). The example defines a low-pass filter, which when applied to an image takes the average of neighboring samples.

```
<FilterSeparable>
  <Filter1D><Terms mpeg7:dim="3"> 1.0 2.0 1.0 </Terms></Filter1D>
  <Filter1D><Terms mpeg7:dim="3"> 1.0 2.0 1.0 </Terms></Filter1D>
</FilterSeparable>

<Filter2D>
  <Terms mpeg7:dim="3 3">
    1.0 2.0 1.0
    2.0 4.0 2.0
    1.0 2.0 1.0
  </Terms>
</Filter2D>
```

13.3.4 Filtering datatype

13.3.4.1 Introduction

The Filtering datatype is used to describe how a filter is applied via convolution to an image, audio or video signal by specifying the filter, the padding, and form of convolution in terms of the lead-in or shift on each dimension. The Filtering datatype includes a single Filter, which can be a 1-D, 2-D or multi-dimensional separable filter, and a sequence of elements with padding and shift information for each dimension. The padding is specified in terms of the paddingType, which indicates the type of border replication such as zero padding, border replication, periodic extension, symmetric extension and so forth.

13.3.4.2 Filtering datatype syntax

```
<!-- ##### -->
<!-- Definition of Filtering datatype (13.3.4) -->
<!-- ##### -->

<!-- Definition of Filtering datatype -->
<complexType name="FilteringType">
  <sequence>
    <element name="Filter" type="mpeg7:FilterType"/>
    <element name="PadSize" type="mpeg7:SignalPlaneSampleType" minOccurs="0"/>
    <element name="Shift" type="mpeg7:SignalPlaneSampleType" minOccurs="0"/>
    <element name="CropStart" type="mpeg7:SignalPlaneSampleType" minOccurs="0"/>
    <element name="CropEnd" type="mpeg7:SignalPlaneSampleType" minOccurs="0"/>
  </sequence>
  <attribute name="xPad" type="mpeg7:paddingType" use="optional"
    default="periodic"/>
  <attribute name="yPad" type="mpeg7:paddingType" use="optional"
    default="periodic"/>
  <attribute name="zPad" type="mpeg7:paddingType" use="optional"
    default="periodic"/>
  <attribute name="tPad" type="mpeg7:paddingType" use="optional"
    default="periodic"/>
</complexType>

<!-- Definition of padding datatype -->
<simpleType name="paddingType">
  <union>
    <simpleType>
      <restriction base="NMTOKEN">
        <enumeration value="zero"/>
        <enumeration value="replication"/>
        <enumeration value="periodic"/>
        <enumeration value="symmetric"/>
        <enumeration value="antiSymmetric"/>
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="mpeg7:termReferenceType"/>
    </simpleType>
  </union>
</simpleType>
```

13.3.4.3 Filtering datatype semantics

Semantics of the FilteringType:

Name	Definition
FilteringType	Describes how a filter is applied via convolution to an image, audio or video signal by specifying the filter, the padding, and form of convolution in terms of the lead-in or shift on each dimension.
Filter	Describes the filter kernel that is applied to an image, audio or video signal.
PadSize	Describes the size of the padding applied to each dimension before convolving the image, video or audio signal with the filter (optional). PadSize describes the padding on the x, y, z, t dimensions as an integer number of samples. The specification of PadSize is optional. If not specified, PadSize shall be assumed to have a value of L-1 on each dimension, where L is the length of the filter on that dimension.
Shift	Describes the lead-in or shift applied to each dimension before convolving the image, video or audio signal with the filter. The Shift refers to the number of coefficients the filter kernel is shifted from the origin at the start of the convolution sum. For example, a shift of zero indicates left-justified convolution form. Shift is specified using SignalPlaneSampleType in order to describe the shift on the x, y, z, t dimensions as an integer number of samples. The specification of Shift is optional. If not specified, Shift shall be assumed to have a value of "0" on each dimension.
CropStart	Describes the cropping of the output of the convolution at the start of the convolution sum. Typically, the convolution expands the signal at the borders by a factor of L-1, where L is the length of the filter. By cropping the output signal at the borders, it can be made the same size in terms of number of samples as the input signal. CropStart specifies the cropping in terms of number of samples that the output signal is cropped at the start of the convolution sum. A value of 0 indicates that cropping was not applied. CropStart is specified using SignalPlaneSampleType in order to describe the cropping on the x, y, z, t dimensions as an integer number of samples. For an image, this refers to the cropping on the left or top of the output image. The specification of CropStart is optional. If not specified, CropStart shall be assumed to have a value of "0" on each dimension.
CropEnd	Describes the cropping of the output of the convolution at the end of the convolution sum. CropEnd specifies the cropping in terms of number of samples that the output signal is cropped at the end of the convolution sum. A value of 0 indicates that cropping was not applied. For an image, CropEnd refers to the cropping on the right or bottom of the output image. The specification of CropEnd is optional. If not specified, CropEnd shall be assumed to have a value of "0" on each dimension.
xPad, yPad, zPad, tPad	Identifies the type of padding applied to each dimension (x, y, z, t) before convolving the image, video or audio signal with the filter (optional). If not specified, padding shall be assumed to have a value of "periodic" on each dimension.

Semantics of the paddingType:

Name	Definition
paddingType	Describes how the border of an image, video or audio signal is extended before convolving the image, video or audio signal with a filter. The different padding types are defined as follows (in the examples, let A, B, C and D denote samples near the start of a signal, and let denote the border of the signal):

Name	Definition
	• <i>zero</i> - Padding with zeros (e.g.: 0 0 A B C D). • <i>replication</i> - Padding with the value nearest to the border (e.g.: A A A B C D). • <i>periodic</i> - Padding using periodic or cyclic extension (e.g.: C D A B C D). • <i>symmetric</i> - Padding using symmetric or mirror extension (e.g.: C B A B C D). • <i>antiSymmetric</i> - Padding using anti-symmetric mirror extension (e.g.: -C -B A B C D). Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.

13.3.4.4 Filtering datatype examples (informative)

The following example describes the filtering of an image using a discrete 2-D filter. The description specifies the padding of the image by one pixel on each dimension using symmetric extension. The description also specifies the shifting of the filter kernel by one sample or pixel in both the horizontal and vertical direction before taking the convolution sum. The description also specifies the output image should be cropped by one pixel at all borders (left, top, right and bottom). The example uses a 2-D filter of size 3 x 3.

```
<Filtering xPad="symmetric" yPad="symmetric" >
  <Filter xsi:type="Filter2DType">
    <Terms mpeg7:dim="3 3">
      1.0 2.0 1.0
      2.0 4.0 2.0
      1.0 2.0 1.0
    </Terms>
  </Filter>
  <PadSize x="1" y="1" />
  <Shift x="1" y="1" />
  <CropStart x="1" y="1" />
  <CropEnd x="1" y="1" />
</Filtering>
```

13.3.5 View DS

13.3.5.1 Introduction

Subclauses 13.3.5 to 13.3.10 describe Space and Frequency Views, which correspond to views of image, video and audio signals in the space and/or frequency domain. The tools specified in subclauses 13.3.5 to 13.3.10 include:

Table 39 — Overview of View Tools

Tools	Functionality
View	This tool describes a view of an image, video or audio signal (abstract).
SpaceView	This tool describes a view of an image, video or audio signal in multi-dimensional space and/or time. For example, a spatial segment of an image can be considered a SpaceView.
FrequencyView	This tool describes a view of an image, video or audio signal in multi-dimensional

Tools	Functionality
	spatial- and/or temporal-frequency. For example, a wavelet subband of an audio signal can be considered a FrequencyView.
SpaceFrequencyView	This tool describes a view of an image, video or audio signal simultaneously in multi-dimensional space and frequency. For example, a wavelet subband of a spatial segment of an image can be considered a SpaceFrequencyView.
ResolutionView	This tool describes a low-resolution view of an image, video or audio signal. For example, a thumbnail view of an image can be considered a ResolutionView. Conceptually, a ResolutionView can be considered as a special case of a FrequencyView that corresponds to a low-frequency subband, where down-sampling shall be used.
SpaceResolutionView	This tool describes a view of an image, video or audio signal simultaneously in space and resolution. For example, a low-resolution view of a spatial segment of an image can be considered a SpaceResolutionView.

The View concepts are illustrated in Figure 71, which shows the different kinds of Space and Frequency Views that can be specified. The examples are illustrated for a 2-D source image, however, the Views can apply equivalently to a source audio (1-D) or video (3-D) signal.

As shown in Figure 71, the SpaceView corresponds to a region of the image. The ResolutionView corresponds to a low-resolution or thumbnail version of the image. The SpaceResolutionView corresponds to a low-resolution version of a region of the image. The FrequencyView corresponds to a frequency subband of the image. For example, the frequency subband may indicate texture or edge information of the image. Finally, the SpaceFrequencyView corresponds to a frequency subband of a region of the image. All of these examples of Views fall under a general category of Space and Frequency Views since they correspond to particular partitions of the image in the space and/or frequency domain.

The View DS provides an abstract type for the specific types of views such as SpaceView, FrequencyView, ResolutionView, SpaceResolutionView, and SpaceFrequencyView. The View DS is used to describe the Source and Target View image, video or audio signal (including their locations).

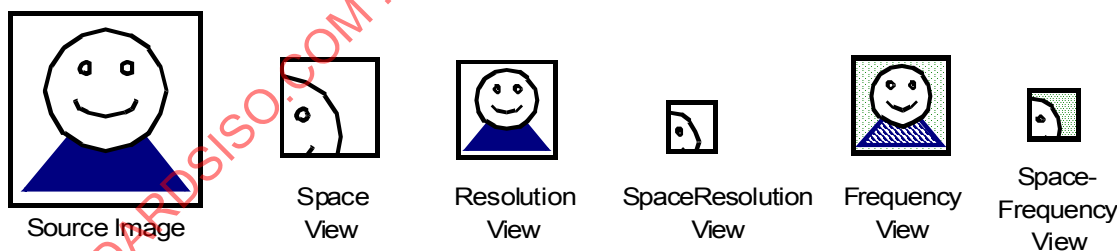


Figure 71 — Illustration of the different types of Space and FrequencyViews of a Source image: SpaceView, ResolutionView, SpaceResolutionView, FrequencyView and SpaceFrequencyView

13.3.5.2 View DS syntax

```
<!-- ##### -->
<!-- Definition of View DS (13.3.5) -->
<!-- ##### -->

<!-- Definition of View DS -->
<complexType name="ViewType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
```

```

        <sequence>
          <element name="Target" type="mpeg7:SignalType"/>
          <element name="Source" type="mpeg7:SignalType" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

```

13.3.5.3 View DS semantics

Semantics of the ViewType:

Name	Definition
ViewType	Describes a view of an image, video, or audio signal (abstract).
Target	Describes the Target image, video or audio signal, which corresponds to the view signal.
Source	Describes the Source image, video or audio signal, which corresponds to the source signal from which the view is derived or defined with respect to (optional). Source is optional since in some cases it is necessary to access only the view signal (described by Target) and not the source signal (described by Source).

13.3.6 SpaceView DS

13.3.6.1 Introduction

The SpaceView DS is used to describe a multi-dimensional view of an image, video or audio signal in space and/or time. In general, a SpaceView is specified in terms of a multi-dimensional Partition in the spatial or temporal domain of the image, video or audio signal. For example, a SpaceView of an image corresponds to a region of the image. A SpaceView of an audio signal corresponds to a temporal segment of the audio signal. A SpaceView of a video may correspond to a spatial region of a video frame or a temporal region of a video segment.

13.3.6.2 SpaceView DS syntax

```

<!-- ##### -->
<!-- Definition of SpaceView DS (13.3.6) -->
<!-- ##### -->

<!-- Definition of SpaceView DS -->
<complexType name="SpaceViewType">
  <complexContent>
    <extension base="mpeg7:ViewType">
      <choice>
        <element name="SpacePartition" type="mpeg7:PartitionType"/>
        <element name="SpaceRegion" type="mpeg7:RegionLocatorType"/>
        <element name="SpatioTemporalRegion"
          type="mpeg7:SpatioTemporalLocatorType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

13.3.6.3 SpaceView DS semantics

Semantics of the SpaceViewType:

Name	Definition
SpaceViewType	Describes a multi-dimensional spatial view of an image, video or audio signal.
SpacePartition	Describes the multi-dimensional rectangle that defines a partition in multi-dimensional spatial or temporal plane spanned by the image, video or audio signal. The SpacePartition is specified in relation to the Source image, video or audio signal. In the case of the SpaceView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the SpaceView, then the SpacePartition is specified in relation to the Source image, video or audio signal of the root of the view decomposition.
SpaceRegion	Describes the multi-dimensional View as a region specified using the RegionLocatorType. SpaceRegion applies only to 2-D image data. It shall not be used otherwise. The RegionLocatorType is defined in ISO/IEC 15938-3.
SpatioTemporalRegion	Describes the multi-dimensional View as a region specified using the SpatioTemporalLocatorType. SpatioTemporalRegion applies only to time-varying 2-D spatial, image sequence or video data. It shall not be used otherwise. The SpatioTemporalLocatorType is defined in ISO/IEC 15938-3.

13.3.6.4 SpaceView DS examples (informative)

The following examples describe different SpaceViews. The first example describes a SpaceView that corresponds to the upper-left quadrant of an image.

```

<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <View xsi:type="SpaceViewType">
      <Target>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://Aerial-upperleft.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Target>
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://Aerial.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <SpacePartition>
        <Origin xOrigin="left" yOrigin="top"/>
        <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
        <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
      </SpacePartition>
    </View>
  </Description>
</Mpeg7>

```


The following example describes a SpaceView that corresponds to the first 16384 samples of an audio signal.

```
<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <View xsi:type="SpaceViewType">
      <Target>
        <AudioSignal>
          <MediaLocator>
            <MediaUri>music-intro.mp3</MediaUri>
          </MediaLocator>
        </AudioSignal>
      </Target>
      <Source>
        <AudioSignal>
          <MediaLocator>
            <MediaUri>music.mp3</MediaUri>
          </MediaLocator>
        </AudioSignal>
      </Source>
      <SpacePartition>
        <Start xsi:type="SignalPlaneSampleType" t="0"/>
        <Extent xsi:type="SignalPlaneSampleType" t="16384"/>
      </SpacePartition>
    </View>
  </Description>
</Mpeg7>
```

13.3.7 FrequencyView DS

13.3.7.1 Introduction

The FrequencyView DS is used to describe a multi-dimensional view of an image, video or audio signal in frequency. In general, a FrequencyView is specified in terms of a multi-dimensional Partition in the spatial-frequency and/or temporal-frequency domain of the image, video or audio signal. For example, a FrequencyView of an image corresponds to a spatial-frequency subband of the image. A FrequencyView of an audio signal corresponds to a temporal-frequency subband of the audio signal. A FrequencyView of a video corresponds to a 3-D subband of the video.

13.3.7.2 FrequencyView DS syntax

```
<!-- ##### -->
<!-- Definition of FrequencyView DS (13.3.7) -->
<!-- ##### -->

<!-- Definition of FrequencyView DS -->
<complexType name="FrequencyViewType">
  <complexContent>
    <extension base="mpeg7:ViewType">
      <sequence>
        <element name="FrequencyPartition" type="mpeg7:PartitionType"/>
        <element name="Filtering" type="mpeg7:FilteringType" minOccurs="0"/>
        <element name="DownSamplingFactor" type="mpeg7:SignalPlaneSampleType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

13.3.7.3 FrequencyView DS semantics

Semantics of the FrequencyViewType:

Name	Definition
FrequencyViewType	Describes a multi-dimensional view of an image, video or audio signal in frequency.
FrequencyPartition	Describes the multi-dimensional rectangle that defines a partition in multi-dimensional spatial-frequency and/or temporal-frequency plane spanned by the image, video or audio signal. The FrequencyPartition is specified in relation to the Source image, video or audio signal. In the case of the FrequencyView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the FrequencyView, then the FrequencyPartition is specified in relation to the Source image, video or audio signal of the root of the view decomposition.
Filtering	Describes how filtering is performed to extract the FrequencyView (optional). The Filtering is defined in terms of a Filter and the padding, shifting and cropping used in processing the Source image, video or audio signal to compute the Target of the view. In part, the Filtering information implicitly characterizes the partition of the multi-dimensional frequency plane in terms of the passband and stopband characteristics of the filter. The specification of Filtering is optional, in which case an ideal filter is assumed, which implies that the partition rectangle has crisp boundaries in the frequency plane. In the case of the FrequencyView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the FrequencyView, then the Filtering is specified in relation to the Source image, video or audio signal of the root of the view decomposition.
DownSamplingFactor	Describes how down-sampling is performed on each dimension in extracting the FrequencyView, with respect to the Source image, video or audio signal (optional). Sampling is specified as a SignalPlaneType that specifies the down-sampling factor, which can take a nonNegativeInteger value greater or equal to one on each dimension. For example, a value of 2 on a dimension indicates that the dimension is down-sampled by two, which means that every other point from the discrete dimension is taken. In the case of the FrequencyView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the FrequencyView, then the DownSamplingFactor is specified in relation to the Source image, video or audio signal of the root of the view decomposition.

13.3.7.4 FrequencyView DS examples (informative)

The following example describes a FrequencyView that corresponds to a twice-iterated, low-pass, low-pass wavelet subband of an audio signal. The FrequencyView corresponds to the first one-quarter of the frequency plane. Furthermore, the filtered signal is down-sampled by a factor of four to produce a Target FrequencyView audio signal that has one-quarter the number of samples as the Source audio signal. The following example describes a coarse version of the audio signal.

```
<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <View xsi:type="FrequencyViewType">
      <Target>
        <AudioSignal>
          <MediaLocator>
            <MediaUri>file://audio-ll.mp3</MediaUri>
          </MediaLocator>
        </AudioSignal>
      </Target>
```

```

    <Source>
      <AudioSignal>
        <MediaLocator>
          <MediaUri>file://audio.mp3</MediaUri>
        </MediaLocator>
      </AudioSignal>
    </Source>
    <FrequencyPartition>
      <Start xsi:type="SignalPlaneFractionType" t="0"/>
      <End xsi:type="SignalPlaneFractionType" t="0.25"/>
    </FrequencyPartition>
    <DownSamplingFactor t="4"/>
  </View>
</Description>
</Mpeg7>

```

13.3.8 SpaceFrequencyView DS

13.3.8.1 Introduction

The SpaceFrequencyView DS is used to describe a multi-dimensional view of an image, video or audio signal in space and frequency. In general, a SpaceFrequencyView is specified in terms of multi-dimensional Partitions in the spatial and/or temporal domain in addition to the spatial-frequency and/or temporal-frequency domain of the image, video or audio signal. For example, a SpaceFrequencyView of an image corresponds to a spatial-frequency subband of a region of the image. A SpaceFrequencyView of an audio signal corresponds to a temporal-frequency subband of a segment of the audio signal. A SpaceFrequencyView of a video corresponds to a 3-D subband of a spatio-temporal region of video.

13.3.8.2 SpaceFrequencyView DS syntax

```

<!-- ##### -->
<!-- Definition of SpaceFrequencyView DS (13.3.8) -->
<!-- ##### -->

<!-- Definition of SpaceFrequencyView DS -->
<complexType name="SpaceFrequencyViewType">
  <complexContent>
    <extension base="mpeg7:ViewType">
      <sequence>
        <choice>
          <element name="SpacePartition" type="mpeg7:PartitionType"/>
          <element name="SpaceRegion" type="mpeg7:RegionLocatorType"/>
        </choice>
        <element name="FrequencyPartition" type="mpeg7:PartitionType"/>
        <element name="Filtering" type="mpeg7:FilteringType" minOccurs="0"/>
        <element name="DownSamplingFactor" type="mpeg7:SignalPlaneSampleType"
          minOccurs="0"/>
      </sequence>
      <attribute name="order" use="optional" default="spaceFrequency">
        <simpleType>
          <restriction base="NMTOKEN">
            <enumeration value="spaceFrequency"/>
            <enumeration value="frequencySpace"/>
            <enumeration value="commutative"/>
            <enumeration value="partitionable"/>
            <enumeration value="mixed"/>
          </restriction>
        </simpleType>
      </attribute>
    </extension>
  </complexContent>
</complexType>

```

```

    </attribute>
  </extension>
</complexContent>
</complexType>

```

13.3.8.3 SpaceFrequencyView DS semantics

Semantics of the SpaceFrequencyViewType:

Name	Definition
SpaceFrequencyViewType	Describes a multi-dimensional view of an image, video or audio signal in space and frequency.
SpacePartition	Describes the multi-dimensional rectangle that defines a partition in multi-dimensional spatial and/or temporal plane spanned by the image, video or audio signal. The SpacePartition is specified in relation to the Source image, video or audio signal. In the case of the SpaceFrequencyView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the SpaceFrequencyView, then the SpacePartition is specified in relation to the Source image, video or audio signal of the root of the view decomposition.
SpaceRegion	Describes the multi-dimensional View as a region specified using the RegionLocatorType. The RegionLocatorType is defined in ISO/IEC 15938-3.
FrequencyPartition	Describes the multi-dimensional rectangle that defines a partition in multi-dimensional spatial-frequency and/or temporal-frequency plane spanned by the image, video or audio signal. The FrequencyPartition is specified in relation to the Source image, video or audio signal. In the case of the SpaceFrequencyView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the SpaceFrequencyView, then the FrequencyPartition is specified in relation to the Source image, video or audio signal of the root of the view decomposition.
Filtering	Describes how filtering is performed to extract the SpaceFrequencyView from the frequency plane (optional). See specification of FrequencyView above for more details.
DownSamplingFactor	Describes how down-sampling is performed on each dimension in extracting the SpaceFrequencyView from the frequency plane, with respect to the Source image, video or audio signal (optional). See specification of FrequencyView above for more details.
order	Identifies the order of the analysis operations in extracting the SpaceFrequencyView from the Source image, video or audio signal. The order shall take values of "spaceFrequency", "frequencySpace", "commutative", "partitionable", or "mixed", which are defined as follows: • <i>spaceFrequency</i> - indicates that the spatial extraction operation is performed before the frequency extraction operation. • <i>frequencySpace</i> - indicates that the frequency extraction operation is performed before the space extraction operation. • <i>commutative</i> - indicates that the extraction involved an unspecified series of space and frequency operations that are commutative, which implies that the order does not matter. An example of a commutative decomposition is one that uses a Haar two-tap filter bank and performs down-sampling by two to perform the frequency extraction. • <i>partitionable</i> - indicates that the extraction involved an unspecified series

Name	Definition
	of space and frequency operations, that are commutative since the filter bank takes a partitionable form, which implies that the order does not matter. An example of a partitionable decomposition is one that uses a twelve-tap filter bank, performs down-sampling by two, and implements the filtering via circular convolution with half the period of the signal to perform the frequency extraction. • <i>mixed</i> - indicates that the extraction involved an unspecified series of space and frequency operations. In this case, it may not be possible to recreate the extraction or perform synthesis using the SpaceFrequencyView data. By default, a value of "spaceFrequency" is assumed for order if not specified.

13.3.8.4 SpaceFrequencyView DS examples (informative)

The following example describes a SpaceFrequencyView that corresponds to a wavelet subband of a region of an image. The SpaceFrequencyView corresponds to the lower one-quarter of the spatial-frequency plane of the upper-left corner region of the image. Furthermore, the filtered region is down-sampled by a factor of two on each dimension to produce a Target SpaceFrequencyView image that has one-sixteenth the number of samples as the Source image. In this example, the order of the space and frequency analysis operations to extract the view is specified as space first then frequency. The following example describes a coarse version of the upper-left region of the image.

```

<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <View xsi:type="SpaceFrequencyViewType" order="spaceFrequency">
      <Target>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://soccer-ul-ll.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Target>
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://soccer.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <SpacePartition>
        <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
        <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
      </SpacePartition>
      <FrequencyPartition>
        <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
        <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
      </FrequencyPartition>
      <DownSamplingFactor x="2" y="2"/>
    </View>
  </Description>
</Mpeg7>

```

13.3.9 ResolutionView DS

13.3.9.1 Introduction

The ResolutionView DS is used to describe a multi-dimensional low-resolution view of an image, video or audio signal. Conceptually, a ResolutionView is a specific kind of FrequencyView that corresponds to the low-frequency portion of the frequency plane, and where the filtered signal has been down-sampled. A ResolutionView is specified in terms of the down-sampling factor on each dimension. Optionally, the ResolutionView can also be described in terms of its frequency partition and filtering. For example, a ResolutionView of an image corresponds to a thumbnail version of the image. A ResolutionView of an audio signal corresponds to a coarse version of the signal.

13.3.9.2 ResolutionView DS syntax

```
<!-- ##### -->
<!-- Definition of ResolutionView DS (13.3.9) -->
<!-- ##### -->

<!-- Definition of ResolutionView DS -->
<complexType name="ResolutionViewType">
  <complexContent>
    <extension base="mpeg7:ViewType">
      <sequence>
        <element name="FrequencyPartition" type="mpeg7:PartitionType"
          minOccurs="0"/>
        <element name="Filtering" type="mpeg7:FilteringType"
          minOccurs="0"/>
        <element name="DownSamplingFactor" type="mpeg7:SignalPlaneSampleType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

13.3.9.3 ResolutionView DS semantics

Semantics of the ResolutionViewType:

Name	Definition
ResolutionViewType	Describes a multi-dimensional low-resolution view of an image, video or audio signal.
FrequencyPartition	Describes the multi-dimensional rectangle that defines a partition in the multi-dimensional spatial-frequency and/or temporal-frequency plane spanned by the image, video or audio signal (optional). The FrequencyPartition is specified in relation to the Source image, video or audio signal. In the case of the ResolutionView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the ResolutionView, then the FrequencyPartition is specified in relation to the Source image, video or audio signal of the root of the view decomposition.
Filtering	Describes how filtering is performed to extract the ResolutionView from the frequency plane (optional). See specification of FrequencyView above for more details.
DownSamplingFactor	Describes how down-sampling is performed on each dimension in extracting the ResolutionView from the frequency plane, with respect to the Source image, video or audio signal. See specification of FrequencyView above for more details.

13.3.9.4 ResolutionView DS examples (informative)

The following example describes a ResolutionView that corresponds to a one-sixteenth size thumbnail view of an image (downsampled by a factor of four along both dimensions).

```
<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <View xsi:type="ResolutionViewType">
      <Target>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://soccer-coarse.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Target>
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://soccer.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <DownSamplingFactor x="4" y="4"/>
    </View>
  </Description>
</Mpeg7>
```

13.3.10 SpaceResolutionView DS

13.3.10.1 Introduction

The SpaceResolutionView DS is used to describe a multi-dimensional view of data, which corresponds to a low-frequency region from the frequency plane.

13.3.10.2 SpaceResolutionView DS syntax

```
<!-- ##### -->
<!-- Definition of SpaceResolutionView DS (13.3.10) -->
<!-- ##### -->

<!-- Definition of SpaceResolutionView DS -->
<complexType name="SpaceResolutionViewType">
  <complexContent>
    <extension base="mpeg7:ViewType">
      <sequence>
        <choice>
          <element name="SpacePartition" type="mpeg7:PartitionType"/>
          <element name="SpaceRegion" type="mpeg7:RegionLocatorType"/>
        </choice>
        <element name="FrequencyPartition" type="mpeg7:PartitionType"
          minOccurs="0"/>
        <element name="Filtering" type="mpeg7:FilteringType"
          minOccurs="0"/>
        <element name="DownSamplingFactor" type="mpeg7:SignalPlaneSampleType"/>
      </sequence>
      <attribute name="order" use="optional" default="spaceFrequency">
        <simpleType>
          <restriction base="NMTOKEN">
```



```

        <enumeration value="spaceFrequency"/>
        <enumeration value="frequencySpace"/>
        <enumeration value="commutative"/>
        <enumeration value="partitionable"/>
        <enumeration value="mixed"/>
    </restriction>
</simpleType>
</attribute>
</extension>
</complexContent>
</complexType>

```

13.3.10.3 SpaceResolutionView DS semantics

Semantics of the SpaceResolutionViewType:

Name	Definition
SpaceResolutionViewType	Describes a multi-dimensional view of an image, video or audio signal in space and resolution.
SpacePartition	Describes the multi-dimensional rectangle that defines a partition in multi-dimensional spatial and/or temporal plane spanned by the image, video or audio signal. The SpacePartition is specified in relation to the Source image, video or audio signal. In the case of the SpaceResolutionView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the SpaceResolutionView, then the SpacePartition is specified in relation to the Source image, video or audio signal of the root of the view decomposition.
SpaceRegion	Describes the multi-dimensional View as a region specified using the RegionLocatorType. The RegionLocatorType is defined in ISO/IEC 15938-3.
FrequencyPartition	Describes the multi-dimensional rectangle that defines a partition in the multi-dimensional spatial-frequency and/or temporal-frequency plane spanned by the image, video or audio signal (optional). The FrequencyPartition is specified in relation to the Source image, video or audio signal. In the case of the SpaceResolutionView being embedded in a view decomposition, if the Source image, video or audio signal is not specified in the SpaceResolutionView, then the FrequencyPartition is specified in relation to the Source image, video or audio signal of the root of the view decomposition.
Filtering	Describes how filtering is performed to extract the SpaceResolutionView from the frequency plane (optional). See specification of FrequencyView above for more details.
DownSamplingFactor	Describes how down-sampling is performed on each dimension in extracting the SpaceResolutionView from the frequency plane, with respect to the Source image, video or audio signal. See specification of FrequencyView above for more details.
order	Identifies the order of the analysis operations in extracting the SpaceResolutionView from the Source image, video or audio signal. See the specification of the SpaceFrequencyView above for more details.

13.3.10.4 SpaceResolutionView DS examples (informative)

The following example describes a SpaceResolutionView that corresponds to a coarse or thumbnail view of a region of an image. The SpaceResolutionView corresponds to the lower one-quarter resolution view of the upper-left corner region of the image. In this example, the order of the space and frequency (resolution) analysis operations to extract the view is specified as space first then frequency (resolution).

```
<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <View xsi:type="SpaceResolutionViewType" order="spaceFrequency">
      <Target>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://soccer-ul-ll.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Target>
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://soccer.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <SpacePartition>
        <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
        <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
      </SpacePartition>
      <DownSamplingFactor x="2" y="2"/>
    </View>
  </Description>
</Mpeg7>
```

13.3.11 ViewDecomposition DS

13.3.11.1 Introduction

Subclauses 13.3.11 to 13.3.17 specify tools for describing different space and frequency organizations and structured decompositions of views of image, video or audio signals. The tools specified in the subclauses 13.3.11 to 13.3.17 include:

Table 40 — Overview of View Decomposition Tools

<i>Tools</i>	<i>Functionality</i>
ViewSet	This tool describes a set of Views, which can have different properties of completeness and redundancy. For example, the set of subbands produced in a wavelet decomposition of an audio signal forms a ViewSet.
SpaceTree	This tool describes a spatial-tree decomposition of the data. For example, a spatial quad-tree image decomposition.
FrequencyTree	This tool describes a frequency-tree decomposition of the data. For example, a wavelet packet-tree image decomposition.
SpaceFrequencyGraph	This tool describes a decomposition of the signal simultaneously in space and frequency in which the views are organized using a space and frequency graph.

<i>Tools</i>	<i>Functionality</i>
VideoViewGraph	This tool describes a decomposition of video signal in both spatial- and/or temporal-frequency. For example, a 3-D subband decomposition of video.
MultiResolutionPyramid	This tool describes a hierarchy of views of image, video or audio signal, for example, an image pyramid.

The ViewDecomposition DS is used to describe a decomposition of an image, video or audio signal into Views. ViewDecomposition is an abstract base class from which specific types of view sets and tree- and graph-based decompositions are derived.

13.3.11.2 ViewDecomposition DS syntax

```

<!-- ##### -->
<!-- Definition of ViewDecomposition DS (13.3.11) -->
<!-- ##### -->

<!-- Definition of ViewDecomposition DS -->
<complexType name="ViewDecompositionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Source" type="mpeg7:SignalType" minOccurs="0"/>
      </sequence>
      <attribute name="complete" type="boolean" use="optional"/>
      <attribute name="nonRedundant" type="boolean" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

13.3.11.3 ViewDecomposition DS semantics

Semantics of the ViewDecompositionType:

<i>Name</i>	<i>Definition</i>
ViewDecompositionType	Describes a decomposition of an image, video or audio signal into Views (abstract).
Source	Describes a Source image, video or audio signal from which the views in the ViewDecomposition are derived or defined with respect to (optional). If the specification of the Source is not given in the individual Views, then the specification of Source in the ViewDecomposition gives the value of the Source for all member Views of the decomposition.
complete	Indicates whether the set of views in the ViewDecomposition completely covers the space and frequency planes (optional). A value of true indicates that the set of views does cover the planes, otherwise a value of false indicates that it does not. If the attribute is not present, then nothing is known about the value of complete.

Name	Definition
nonRedundant	Indicates whether the set of views in the ViewDecomposition nonredundantly covers the space and frequency planes (optional). A value of true indicates that the set of views do not overlap in the space and frequency planes, otherwise a value of false indicates that set of views is redundant, or that there is overlap in the space or frequency plane. If the attribute is not present, then nothing is known about the value of nonRedundant.

13.3.12 ViewSet DS

13.3.12.1 Introduction

A ViewSet describes a set of Views, which has properties indicating completeness and redundancy in covering the space and/or frequency planes. A ViewSet is complete if it completely covers the space and frequency planes. A ViewSet is nonredundant if the Views do not overlap in the space and frequency planes. Additionally, specific ViewSets can be characterized by the types of views in the set. For example, a ViewSet may contain a set of spatial regions or segments from an image or video. Another type of ViewSet may contain a set of frequency subbands, which may correspond to a uniform set or wavelet, or octave-spaced, set of subbands.

13.3.12.2 ViewSet DS syntax

```

<!-- ##### -->
<!-- Definition of ViewSet DS (13.3.12) -->
<!-- ##### -->

<!-- Definition of ViewSet DS -->
<complexType name="ViewSetType">
  <complexContent>
    <extension base="mpeg7:ViewDecompositionType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="View" type="mpeg7:ViewType"/>
        <element name="ViewRef" type="mpeg7:ReferenceType"/>
      </choice>
      <attribute name="setProperty" use="optional">
        <simpleType>
          <union>
            <simpleType>
              <restriction base="NMTOKEN">
                <enumeration value="space"/>
                <enumeration value="frequency"/>
                <enumeration value="subband"/>
                <enumeration value="uniformSubband"/>
                <enumeration value="wavelet"/>
              </restriction>
            </simpleType>
            <simpleType>
              <restriction base="mpeg7:termReferenceType"/>
            </simpleType>
          </union>
        </simpleType>
      </attribute>
    </extension>
  </complexContent>
</complexType>

```

13.3.12.3 ViewSet DS semantics

Semantics of the ViewSetType:

Name	Definition
ViewSetType	Describes set of views of image, video or audio signals.
View	Describes a view that is a member of the ViewSet (unbounded). View has type ViewType.
ViewRef	Describes a reference to a View that is a member of the ViewSet (unbounded).
setProperty	A property of the set of views, for example, whether they consist exclusively of tiles of the space or frequency plane, or correspond to particular types of sets (optional). The setProperty attribute shall take the values of "space", "frequency", "subband", "uniformSubband", "wavelet", which are defined as follows: ▪ <i>space</i> - indicates that the set of views consists exclusively of SpaceViews. ▪ <i>frequency</i> - indicates that the set of views consists exclusively of FrequencyViews. ▪ <i>subband</i> - indicates that the set of views consists exclusively of FrequencyViews that have been maximally down-sampled. ▪ <i>uniformSubband</i> - indicates that the set of views consists exclusively of subbands of uniform size and spacing in the frequency plane. ▪ <i>wavelet</i> - indicates that the set of views consists exclusively of subbands of octavely decreasing sizing and spacing with lower frequencies in the frequency plane. Other values that are datatype-valid with respect to mpeg7:termReferenceType are reserved.

13.3.12.4 ViewSet DS examples (informative)

The following example describes a ViewSet that contains two SpaceViews. The ViewSet is specified to be nonredundant and not complete. In this example, the two views correspond to the upper-left and upper-right quadrants, respectively, of the source image specified as the source of the view set. In both cases, the specification of the SpacePartition is given in relation to the source image of the view set.

```

<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <ViewSet complete="false" nonRedundant="true" setProperty="space">
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>file://soccer.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <View xsi:type="SpaceViewType">
        <Target>
          <ImageSignal>
            <MediaLocator>
              <MediaUri>file://soccer-upper-left.jpg</MediaUri>
            </MediaLocator>
          </ImageSignal>
        </Target>
      </View>
    </ViewSet>
  </Description>
</Mpeg7>

```

```

        </ImageSignal>
    </Target>
    <SpacePartition>
        <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
        <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
    </SpacePartition>
</View>
<View xsi:type="SpaceViewType">
    <Target>
        <ImageSignal>
            <MediaLocator>
                <MediaUri>file://soccer-upper-right.jpg </MediaUri>
            </MediaLocator>
        </ImageSignal>
    </Target>
    <SpacePartition>
        <Start xsi:type="SignalPlaneFractionType" x="0.5" y="0"/>
        <End xsi:type="SignalPlaneFractionType" x="1.0" y="0.5"/>
    </SpacePartition>
</View>
</ViewSet>
</Description>
</Mpeg7>

```

13.3.13 SpaceTree DS

13.3.13.1 Introduction

A SpaceTree describes a tree-based spatial decomposition of an image, video or audio signal. A SpaceTree consists of SpaceTree nodes, which can optionally contain a SpaceView. Each node also has a number of children, which are also SpaceTrees. The branching factor of the SpaceTree specifies the branching of each node into a number of children. For example, an image spatial quad-tree is an instance of a SpaceTree in which the branching factor has a value of four.

13.3.13.2 SpaceTree DS syntax

```

<!-- ##### -->
<!-- Definition of SpaceTree DS (13.3.13) -->
<!-- ##### -->

<!-- Definition of SpaceTree DS -->
<complexType name="SpaceTreeType">
    <complexContent>
        <extension base="mpeg7:ViewDecompositionType">
            <sequence>
                <choice minOccurs="0">
                    <element name="View" type="mpeg7:SpaceViewType"/>
                    <element name="ViewRef" type="mpeg7:ReferenceType"/>
                </choice>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="Child" type="mpeg7:SpaceTreeType"/>
                    <element name="ChildRef" type="mpeg7:ReferenceType"/>
                </choice>
            </sequence>
            <attribute name="branching" type="nonNegativeInteger" use="required"/>
        </extension>
    </complexContent>
</complexType>

```

13.3.13.3 SpaceTree DS semantics

Semantics of the SpaceTreeType:

<i>Name</i>	<i>Definition</i>
SpaceTreeType	Describes a tree-based spatial decomposition of an image, video or audio signal, such as a spatial quad-tree image decomposition.
View	Describes a View that is contained within the node of the SpaceTree (optional).
ViewRef	Describes a reference to a View that is contained within the node of the SpaceTree (optional).
Child	Describes the SpaceTrees that are descendents of the current node in the SpaceTree (optional). Generally, the number of children follows the factor specified by the value of branching. The exception being the terminal nodes of the SpaceTree, which have no children.
ChildRef	Describes the references to the SpaceTrees that are descendents of the current node in the SpaceTree (optional).
branching	Indicates the branching factor of the current node of the SpaceTree into the number of children SpaceTrees. The total number of Child and ChildRef elements at the current node of the SpaceTree is equal to the branching factor. In the case of terminal nodes, a value of 0 shall be used.

13.3.13.4 SpaceTree DS examples (informative)

The following example describes a SpaceTree with branching factor of two at the root node, which corresponds to the splitting of the source image into two child space views. The specification of the SpacePartitions of the Child views is given in relation to the Source image of the SpaceTree.

```

<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <ViewDecomposition xsi:type="SpaceTreeType" complete="false"
      nonRedundant="true" branching="2">
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>soccer.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <Child branching="0">
        <View>
          <Target>
            <ImageSignal>
              <MediaLocator>
                <MediaUri>soccer-00.jpg</MediaUri>
              </MediaLocator>
            </ImageSignal>
          </Target>
          <SpacePartition>
            <Start xsi:type="SignalPlaneFractionType" x="0.0" y="0"/>

```

```

        <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
      </SpacePartition>
    </View>
  </Child>
  <Child branching="0">
    <View>
      <Target>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>soccer-01.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Target>
      <SpacePartition>
        <Start xsi:type="SignalPlaneFractionType" x="0.5" y="0"/>
        <End xsi:type="SignalPlaneFractionType" x="1.0" y="0.5"/>
      </SpacePartition>
    </View>
  </Child>
</ViewDecomposition>
</Description>
</Mpeg7>

```

13.3.14 FrequencyTree DS

13.3.14.1 Introduction

A FrequencyTree describes a tree-based decomposition of an image, video or audio signal in frequency. A FrequencyTree consists of FrequencyTree nodes, which can optionally contain a FrequencyView. Each node also has a number of children, which are also FrequencyTrees. The branching factor of the FrequencyTree specifies the branching of each node into a number of children. For example, an image wavelet packet-tree is an instance of a FrequencyTree in which the branching factor has a value of four.

13.3.14.2 FrequencyTree DS syntax

```

<!-- ##### -->
<!-- Definition of FrequencyTree DS (13.3.14) -->
<!-- ##### -->

<!-- Definition of FrequencyTree DS -->
<complexType name="FrequencyTreeType">
  <complexContent>
    <extension base="mpeg7:ViewDecompositionType">
      <sequence>
        <choice minOccurs="0">
          <element name="View" type="mpeg7:FrequencyViewType"/>
          <element name="ViewRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Child" type="mpeg7:FrequencyTreeType"/>
          <element name="ChildRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
      <attribute name="branching" type="nonNegativeInteger" use="required"/>
    </extension>
  </complexContent>
</complexType>

```

13.3.14.3 FrequencyTree DS semantics

Semantics of the FrequencyTreeType:

<i>Name</i>	<i>Definition</i>
FrequencyTreeType	Describes a tree-based decomposition of an image, video or audio signal in frequency, such as a wavelet packet-tree decomposition of an image.
View	Describes a View that is contained within the node of the FrequencyTree (optional).
ViewRef	Describes a reference to a View that is contained within the node of the FrequencyTree (optional).
Child	Describes the FrequencyTrees that are descendents of the current node in the FrequencyTree (optional). Generally, the number of children follows the factor specified by the value of branching. The exception being the terminal nodes of the FrequencyTree, which have no children.
ChildRef	Describes the references to the FrequencyTrees that are descendents of the current node in the FrequencyTree (optional).
branching	Indicates the branching factor of the current node of the FrequencyTree into the number of children FrequencyTrees. The total number of Child and ChildRef elements at the current node of the FrequencyTree is equal to the branching factor. In the case of terminal nodes, a value of 0 shall be used.

13.3.14.4 FrequencyTree DS examples (informative)

The following example describes a FrequencyTree with branching factor of two at the root node. The FrequencyTree specifies one FrequencyView at the first level of the tree and two FrequencyView at the second level of the tree.

```
<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <ViewDecomposition xsi:type="FrequencyTreeType" complete="false"
      nonRedundant="true" branching="2">
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>soccer.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <Child branching="0">
        <View>
          <Target>
            <ImageSignal>
              <MediaLocator>
                <MediaUri>soccer-11.jpg</MediaUri>
              </MediaLocator>
            </ImageSignal>
          </Target>
          <FrequencyPartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
            <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
          </FrequencyPartition>
        </View>
      </Child>
    </ViewDecomposition>
  </Description>
</Mpeg7>
```



```

    </View>
  </Child>
  <Child branching="0">
    <View>
      <Target>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>soccer-lh.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Target>
      <FrequencyPartition>
        <Start xsi:type="SignalPlaneFractionType" x="0" y="0.5"/>
        <End xsi:type="SignalPlaneFractionType" x="0.5" y="1"/>
      </FrequencyPartition>
    </View>
  </Child>
</ViewDecomposition>
</Description>
</Mpeg7>

```

13.3.15 SpaceFrequencyGraph DS

13.3.15.1 Introduction

A SpaceFrequencyGraph describes a directed acyclic graph-based decomposition of an image, video or audio signal in space and frequency. A SpaceFrequencyGraph consists of SpaceFrequencyGraph nodes, which can optionally contain a SpaceFrequencyView. Each node also has a number of children in the space and frequency dimensions, which are also SpaceFrequencyGraphs. The branching factor of the SpaceFrequencyGraph specifies the branching of each node into a number of children on the space and frequency dimensions.

13.3.15.2 SpaceFrequencyGraph DS syntax

```

<!-- ##### -->
<!-- Definition of SpaceFrequencyGraph DS (13.3.15) -->
<!-- ##### -->

<!-- Definition of SpaceFrequencyGraph DS -->
<complexType name="SpaceFrequencyGraphType">
  <complexContent>
    <extension base="mpeg7:ViewDecompositionType">
      <sequence>
        <choice minOccurs="0">
          <element name="View" type="mpeg7:SpaceFrequencyViewType"/>
          <element name="ViewRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SpaceChild"
            type="mpeg7:SpaceFrequencyGraphType"/>
          <element name="SpaceChildRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="FrequencyChild"
            type="mpeg7:SpaceFrequencyGraphType"/>
          <element name="FrequencyChildRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
      <attribute name="branching" type="nonNegativeInteger" use="required"/>
    </extension>
  </complexContent>
</complexType>

```

```

</extension>
</complexContent>
</complexType>

```

13.3.15.3 SpaceFrequencyGraph DS semantics

Semantics of the SpaceFrequencyGraphType:

Name	Definition
SpaceFrequencyGraphType	Describes a directed acyclic graph-based decomposition of an image, video or audio signal in space and frequency.
View	Describes a View that is contained within the node of the SpaceFrequencyGraph (optional).
ViewRef	Describes a reference to a View that is contained within the node of the SpaceFrequencyGraph (optional).
SpaceChild	Describes the SpaceFrequencyGraphs that are descendents of the current node in the SpaceFrequencyGraph in the spatial dimension. Generally, the number of children follows the factor specified by the value of branching. The exception being the terminal nodes of the SpaceFrequencyGraph, which have no children.
SpaceChildRef	Describes the references to the SpaceFrequencyGraphs that are descendents of the current node in the SpaceFrequencyGraph in the spatial dimension. The references allow the formation of a directed-acyclic graph (DAG) from the hierarchical tree decomposition. The specification allows the choice of embedding the children directly using SpaceChild or including links to the space children via SpaceChildRef.
FrequencyChild	Describes the SpaceFrequencyGraphs that are descendents of the current node in the SpaceFrequencyGraph in the frequency dimension. Generally, the number of children follows the factor specified by the value of branching. The exception being the terminal nodes of the SpaceFrequencyGraph, which have no children.
FrequencyChildRef	Describes the references to the SpaceFrequencyGraphs that are descendents of the current node in the SpaceFrequencyGraph in the frequency dimension. The references allow the formation of a directed-acyclic graph (DAG) from the hierarchical tree decomposition. The specification allows the choice of embedding the children directly using FrequencyChild or including links to the frequency children via FrequencyChildRef.
branching	Indicates the branching factor of the current node of the SpaceFrequencyGraph into the number of children SpaceFrequencyGraphs along both the space and frequency dimensions. At the current node of the SpaceFrequencyGraph, the total number of SpaceChild and SpaceChildRef elements, and the total number of FrequencyChild and FrequencyChildRef elements, is equal to the branching factor. In the case of terminal nodes, a value of 0 shall be used.

13.3.15.4 SpaceFrequencyGraph DS examples (informative)

The following example describes a SpaceFrequencyGraph with branching factor of two on each space and frequency dimension at the root node. The SpaceFrequencyGraph specifies one SpaceFrequencyView at the first level of the directed acyclic graph and four SpaceFrequencyViews at the second level of the graph.

```

<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <ViewDecomposition xsi:type="SpaceFrequencyGraphType" complete="false"
      nonRedundant="false" branching="2">
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>soccer.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <SpaceChild branching="0">
        <View>
          <Target>
            <ImageSignal>
              <MediaLocator>
                <MediaUri>soccer-0.jpg</MediaUri>
              </MediaLocator>
            </ImageSignal>
          </Target>
          <SpacePartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
            <End xsi:type="SignalPlaneFractionType" x="0.5" y="1"/>
          </SpacePartition>
          <FrequencyPartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
            <End xsi:type="SignalPlaneFractionType" x="1" y="1"/>
          </FrequencyPartition>
        </View>
      </SpaceChild>
      <SpaceChild branching="0">
        <View>
          <Target>
            <ImageSignal>
              <MediaLocator>
                <MediaUri>soccer-1.jpg</MediaUri>
              </MediaLocator>
            </ImageSignal>
          </Target>
          <SpacePartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
            <End xsi:type="SignalPlaneFractionType" x="1" y="0.5"/>
          </SpacePartition>
          <FrequencyPartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
            <End xsi:type="SignalPlaneFractionType" x="1" y="1"/>
          </FrequencyPartition>
        </View>
      </SpaceChild>
      <FrequencyChild branching="0">
        <View>
          <Target>
            <ImageSignal>
              <MediaLocator>
                <MediaUri>soccer-1l.jpg</MediaUri>
              </MediaLocator>
            </ImageSignal>
          </Target>
          <SpacePartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
            <End xsi:type="SignalPlaneFractionType" x="1" y="1"/>
          </SpacePartition>
        </View>
      </FrequencyChild>
    </ViewDecomposition>
  </Description>
</Mpeg7>

```

```

        </SpacePartition>
        <FrequencyPartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
            <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5"/>
        </FrequencyPartition>
    </View>
</FrequencyChild>
<FrequencyChild branching="0">
    <View>
        <Target>
            <ImageSignal>
                <MediaLocator>
                    <MediaUri>soccer-lh.jpg</MediaUri>
                </MediaLocator>
            </ImageSignal>
        </Target>
        <SpacePartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0"/>
            <End xsi:type="SignalPlaneFractionType" x="1" y="1"/>
        </SpacePartition>
        <FrequencyPartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0.5"/>
            <End xsi:type="SignalPlaneFractionType" x="0.5" y="1"/>
        </FrequencyPartition>
    </View>
</FrequencyChild>
</ViewDecomposition>
</Description>
</Mpeg7>

```

13.3.16 VideoViewGraph DS

13.3.16.1 Introduction

A VideoViewGraph describes a directed acyclic graph-based decomposition of an image, video or audio signal in spatial- and/or temporal-frequency dimensions. A VideoViewGraph consists of VideoViewGraph nodes, which can optionally contain a FrequencyView. Each node also has a number of children on the spatial- and/or temporal-frequency dimensions, which are also VideoViewGraphs. The branching factor of the VideoViewGraph specifies the branching of each node into a number of children in the spatial- and/or temporal-frequency dimensions.

13.3.16.2 VideoViewGraph DS syntax

```

<!-- ##### -->
<!-- Definition of VideoViewGraph DS (13.3.16) -->
<!-- ##### -->

<!-- Definition of VideoViewGraph DS -->
<complexType name="VideoViewGraphType">
    <complexContent>
        <extension base="mpeg7:ViewDecompositionType">
            <sequence>
                <choice minOccurs="0">
                    <element name="View" type="mpeg7:FrequencyViewType"/>
                    <element name="ViewRef" type="mpeg7:ReferenceType"/>
                </choice>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="SpaceFrequencyChild"

```

```

        type="mpeg7:VideoViewGraphType"/>
        <element name="SpaceFrequencyChildRef"
            type="mpeg7:ReferenceType"/>
    </choice>
    <choice minOccurs="0" maxOccurs="unbounded">
        <element name="TimeFrequencyChild"
            type="mpeg7:VideoViewGraphType"/>
        <element name="TimeFrequencyChildRef"
            type="mpeg7:ReferenceType"/>
    </choice>
</sequence>
<attribute name="branching" type="nonNegativeInteger" use="required"/>
</extension>
</complexContent>
</complexType>

```

13.3.16.3 VideoViewGraph DS semantics

Semantics of the VideoViewGraphType:

Name	Definition
VideoViewGraphType	Describes a directed acyclic graph-based decomposition of an image, video or audio signal in spatial- and/or temporal-frequency.
View	Describes a View that is contained within the node of the VideoViewGraph (optional).
ViewRef	Describes a reference to a View that is contained within the node of the VideoViewGraph (optional).
SpaceFrequencyChild	Describes the VideoViewGraphs that are descendents of the current node in the VideoViewGraph in the spatial-frequency dimension. Generally, the number of children follows the factor specified by the value of branching. The exception being the terminal nodes of the VideoViewGraph, which have no children.
SpaceFrequencyChildRef	Describe the reference to the VideoViewGraphs that are descendents of the current node in the VideoViewGraph in the spatial-frequency dimension. The links allow the formation of a directed-acyclic graph (DAG) from the hierarchical tree decomposition. The specification allows the choice of embedding the children directly using SpaceFrequencyChild or including links to the spatial-frequency children via SpaceFrequencyChildRef.
TimeFrequencyChild	Describes the VideoViewGraphs that are descendents of the current node in the VideoViewGraph in the temporal-frequency dimension. Generally, the number of children follows the factor specified by the value of branching. The exception being the terminal nodes of the VideoViewGraph, which have no children.
TimeFrequencyChildRef	Describes the links to the VideoViewGraphs that are descendents of the current node in the VideoViewGraph in the temporal-frequency dimension. The links allow the formation of a directed-acyclic graph (DAG) from the hierarchical tree decomposition. The specification allows the choice of embedding the children directly using TimeFrequencyChildRef or including links to the temporal-frequency children via TimeFrequencyChildRef.

Name	Definition
branching	Indicates the branching factor of the current node of the VideoViewGraph into the number children VideoViewGraphs along both the spatial- and/or temporal-frequency dimensions. At the current node of the VideoViewGraph, the total number of SpaceFrequencyChild and SpaceFrequencyChildRef elements, and the total number of TimeFrequencyChild and TimeFrequencyChildRef elements, is equal to the branching factor. In the case of terminal nodes, a value of 0 shall be used.

13.3.16.4 VideoViewGraph DS examples (informative)

The following example describes a VideoViewGraph with branching factor of two on each spatial- and/or temporal-frequency dimension at the root node. The VideoViewGraph specifies one FrequencyView at the first level of the directed acyclic graph and four FrequencyViews at the second level of the graph.

```
<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <ViewDecomposition xsi:type="VideoViewGraphType" complete="true"
      nonRedundant="false" branching="2">
      <Source>
        <VideoSignal>
          <MediaLocator>
            <MediaUri>file://soccer.mpg</MediaUri>
          </MediaLocator>
        </VideoSignal>
      </Source>
      <SpaceFrequencyChild complete="false" nonRedundant="true" branching="0">
        <View>
          <Target>
            <VideoSignal>
              <MediaLocator>
                <MediaUri>soccer-s0.mpg</MediaUri>
              </MediaLocator>
            </VideoSignal>
          </Target>
          <FrequencyPartition>
            <Start xsi:type="SignalPlaneFractionType" x="0" y="0" t="0"/>
            <End xsi:type="SignalPlaneFractionType" x="0.5" y="0.5" t="0.5"/>
          </FrequencyPartition>
        </View>
      </SpaceFrequencyChild>
      <SpaceFrequencyChild complete="false" nonRedundant="true" branching="0">
        <View>
          <Target>
            <VideoSignal>
              <MediaLocator>
                <MediaUri>soccer-s1.mpg</MediaUri>
              </MediaLocator>
            </VideoSignal>
          </Target>
          <FrequencyPartition>
            <Start xsi:type="SignalPlaneFractionType" x="0.5" y="0.5" t="0"/>
            <End xsi:type="SignalPlaneFractionType" x="1" y="1" t="1"/>
          </FrequencyPartition>
        </View>
      </SpaceFrequencyChild>
      <TimeFrequencyChild complete="false" nonRedundant="true" branching="0">
```

```

    <View>
      <Target>
        <VideoSignal>
          <MediaLocator>
            <MediaUri>soccer-t0.mpg</MediaUri>
          </MediaLocator>
        </VideoSignal>
      </Target>
      <FrequencyPartition>
        <Start xsi:type="SignalPlaneFractionType" x="0" y="0" t="0"/>
        <End xsi:type="SignalPlaneFractionType" x="1" y="1" t="0.5"/>
      </FrequencyPartition>
    </View>
  </TimeFrequencyChild>
  <TimeFrequencyChild complete="false" nonRedundant="true" branching="0">
    <View>
      <Target>
        <VideoSignal>
          <MediaLocator>
            <MediaUri>soccer-t1.mpg</MediaUri>
          </MediaLocator>
        </VideoSignal>
      </Target>
      <FrequencyPartition>
        <Start xsi:type="SignalPlaneFractionType" x="0" y="0" t="0.5"/>
        <End xsi:type="SignalPlaneFractionType" x="1" y="1" t="1"/>
      </FrequencyPartition>
    </View>
  </TimeFrequencyChild>
</ViewDecomposition>
</Description>
</Mpeg7>

```

13.3.17 MultiResolutionPyramid DS

13.3.17.1 Introduction

A MultiResolutionPyramid describes a hierarchy of views of image, video or audio signal with differing resolutions.

13.3.17.2 MultiResolutionPyramid DS syntax

```

<!-- ##### -->
<!-- Definition of MultiResolutionPyramid DS (13.3.17) -->
<!-- ##### -->

<!-- Definition of MultiResolutionPyramid DS -->
<complexType name="MultiResolutionPyramidType">
  <complexContent>
    <extension base="mpeg7:ViewDecompositionType">
      <sequence>
        <choice minOccurs="0">
          <element name="View" type="mpeg7:ResolutionViewType"/>
          <element name="ViewRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0">
          <element name="Child" type="mpeg7:MultiResolutionPyramidType"/>
          <element name="ChildRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
      <attribute name="level" type="nonNegativeInteger" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

13.3.17.3 MultiResolutionPyramid DS semantics

Semantics of the MultiResolutionPyramidType:

Name	Definition
MultiResolutionPyramidType	Describes a hierarchical multi-resolution decomposition of an image, video or audio signal.
View	Describes a View that is contained within the node of the MultiResolutionPyramid (optional).
ViewRef	Describes a reference to a View that is contained within the node of the MultiResolutionPyramid (optional).
Child	Describes the MultiResolutionPyramids that are descendents of the current node in the MultiResolutionPyramid (optional).
ChildRef	Describes a reference to MultiResolutionPyramids that are descendents of the current node in the MultiResolutionPyramid (optional).
level	Identifies the level of the current node in the MultiResolutionPyramid as the distance from the root or the hierarchy (optional).

13.3.17.4 MultiResolutionPyramid DS examples (informative)

The following example describes a two-level MultiResolutionPyramid consisting of a Gaussian pyramid of image views.


```

<Mpeg7>
  <Description xsi:type="ViewDescriptionType">
    <ViewDecomposition xsi:type="MultiResolutionPyramidType" level="1">
      <Source>
        <ImageSignal>
          <MediaLocator>
            <MediaUri>soccer.jpg</MediaUri>
          </MediaLocator>
        </ImageSignal>
      </Source>
      <Child level="1">
        <View>
          <Target>
            <ImageSignal>
              <MediaLocator>
                <MediaUri>soccer-1.jpg</MediaUri>
              </MediaLocator>
            </ImageSignal>
          </Target>
          <DownSamplingFactor x="2" y="2"/>
        </View>
        <Child level="2">
          <View>
            <Target>
              <ImageSignal>
                <MediaLocator>
                  <MediaUri>soccer-2.jpg</MediaUri>
                </MediaLocator>
              </ImageSignal>
            </Target>
            <DownSamplingFactor x="4" y="4"/>
          </View>
        </Child>
      </Child>
    </ViewDecomposition>
  </Description>
</Mpeg7>

```

13.4 Variations of the content

13.4.1 Introduction

Subclause 13.4 specifies a tool for describing the associations or relationships between different variations of multimedia content. The Variation DS supports content management by tracking the variations of multimedia content that result from various types of multimedia processing such as summarization, translation, reduction, revision, transcoding and so forth. The Variation DS also supports Universal Multimedia Access by allowing the selection of the most appropriate variation of the multimedia content for the specific capabilities of the terminal devices, network conditions or user preferences.

The key concepts used in the Variation DS are the concepts of VariationSet, Variation, fidelity, priority, timeOffset, timeScale and VariationRelationship, as illustrated above. A VariationSet describes a set of variations of a multimedia content item. The VariationSet refers to the source content using the MultimediaContent DS. The MultimediaContent DS allows the description of MediaInformation, CreationInformation and UsageInformation for the source multimedia content. In particular, MediaInformation may include MediaFormat information, which may be useful for selecting among the variations for Universal Multimedia Access. The VariationSet includes an unbounded set of Variation elements, which allows multiple variations of a single source content to be described.

The Variation refers to the variation multimedia content using the MultimediaContent DS. In addition, the Variation includes information specifying the fidelity, priority, time offset, time scale and relationship type of the variation multimedia content. The fidelity attribute indicates the fidelity of the variation content with respect to the source content, i.e., it expresses how "close" or "faithful" the variation is to the original. The priority attribute indicates the priority of the variation content with respect to the other Variations that may be included in a VariationSet. The timeOffset attribute indicates an offset of the time-base of the variation content with respect to the source content and the timeScale attribute indicates a scaling of the time-line of the variation content with respect to the source content.

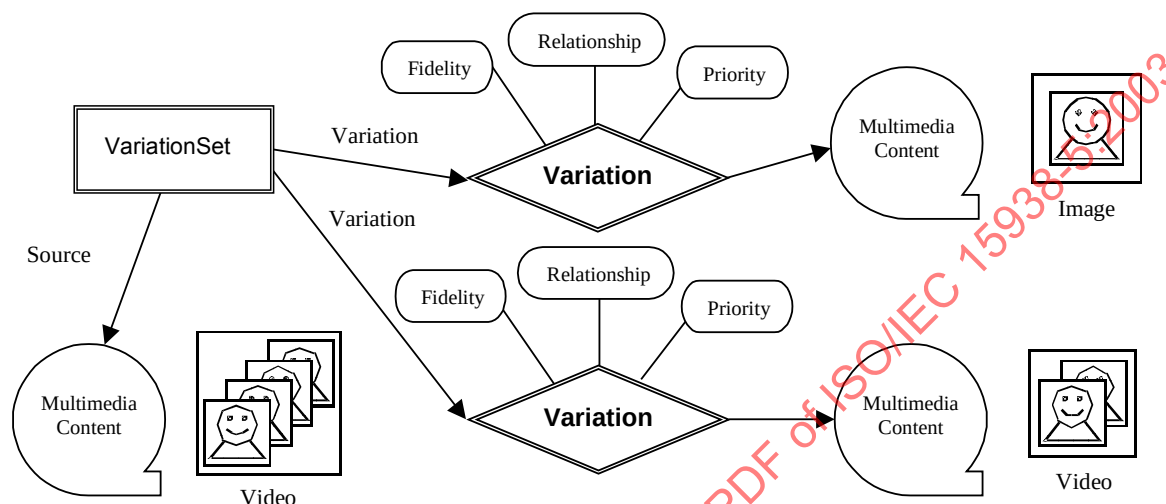


Figure 72 — Illustration of key concepts of Variation DS including VariationSet, Variation, variationFidelity, variationPriority and VariationRelationship

The VariationRelationship attribute specifies the type of association or relationship of the variation multimedia content with the source content. The different types of VariationRelationships are given as follows:

- Summarization - summarizes the important information of the source content in order to support efficient browsing and navigation. Examples of summaries include those defined in the Summary DS such as hierarchical summaries and sequential summaries.
- Abstraction - presents the salient points of a multimedia content. An abstract differs from a summary in that the abstract is separately authored while a summary is derived from the multimedia content.
- Extraction - extracts multimedia content from a multimedia content. Example extracts include key frames selected from video, audio-band and voice excerpts from audio content, paragraphs and key terms from text, and regions, segments, objects, and events from audio and video programs.
- ModalityTranslation - involves the conversion from one multimedia content modality (image, video, text, audio, graphics) to another. Examples of modality translation include text-to-speech (TTS), speech-to-text (speech recognition), video-to-image (video mosaicing), image-to-text (video-text or embedded caption recognition), and graphics-to-image (graphics rendering).
- LanguageTranslation - involves the conversion of text or speech associated with a multimedia content from one language to another, such as Spanish to English.
- ColorReduction - involves the reduction of the color detail of visual content such as an image or video. Examples include the reduction of a 24-bit RGB color image to 8-bit grayscale.

- **SpatialReduction** - involves the reduction in the spatial size of visual content such as an image or video. Examples include the spatial reduction of frames in a video from spatial size 640x480 to 320x240 (width x height).
- **TemporalReduction** - involves the reduction in the time duration of audio or video content. Examples include the temporal reduction of a video from 1 hour to 15 minutes.
- **SamplingReduction** - involves the reduction in the temporal sampling period of audio or video content without shortening of the overall playback time. Examples include the frame rate reduction of video from 30 frames/sec to 15 frames/sec.
- **RateReduction** - involves the reduction in the temporal data rate of audio or video content. Examples include the rate reduction of MP3 audio from 128 Kbps to 96 Kbps.
- **QualityReduction** - involves the reduction in the detail or perceived quality of the multimedia content. Examples include the reduction of quality associated with the quantization of images from 256 levels of intensity to 32 levels of intensity.
- **Compression** - involves the reduction in the amount of data used in representing multimedia content. Examples include the lossy compression of a 1MB image to 256KB.
- **Scaling** - involves selecting a representation of a multimedia content from one of the levels in a scalable data representation. Examples include the scaled representation of an image from the coarse level of a multi-resolution image pyramid.
- **Revision** - indicates that a multimedia content has been revised in some way, such as through editing or post-processing, to produce the variation.
- **Substitution** - indicates that one multimedia content can be used as a substitute for another, without specifying any other explicit relationship between the programs. Examples of substitution include "alt" text in HTML, which is text that can substitute for an image that cannot be handled by the terminal device, or an audio track that replaces a chart in a presentation.
- **Replay** - indicates that the variation content refers to a replay such as a slow-motion replay of the source content.
- **AlternativeView** - indicates that the variation content provides an alternative view of the source content such as a view from another camera.
- **AlternativeMediaProfile** - indicates that the Variation refers to an alternative media profile of the source content.

13.4.2 VariationSet DS

13.4.2.1 Introduction

The Variation DSs are used to describe different variations of multimedia content. The variations may, in general, be generated in a number of different ways, or reflect revisions of the source content. The VariationSet DS specifies the source content and a set of its Variations. Each Variation indicates the variation content in addition to the fidelity to the source, the priority amongst the set of variations and the relationship of the variation with the source content.

13.4.2.2 VariationSet DS syntax

```
<!-- ##### -->
<!-- Definition of VariationSet DS (13.4.2) -->
<!-- ##### -->

<!-- Definition of VariationSet DS -->
<complexType name="VariationSetType">
  <complexContent>
    <extension base="mpeg7:DSType">
```

```

    <sequence>
      <element name="Source" type="mpeg7:MultimediaContentType"/>
      <element name="Variation" type="mpeg7:VariationType"
        minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </extension>
</complexContent>
</complexType>

<!-- Definition of Variation content entity -->
<complexType name="VariationType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <element name="Source" type="mpeg7:MultimediaContentType"
          minOccurs="0"/>
        <element name="Content" type="mpeg7:MultimediaContentType"/>
        <element name="VariationRelationship" minOccurs="0"
          maxOccurs="unbounded">
          <simpleType>
            <union>
              <simpleType>
                <restriction base="NMTOKEN">
                  <enumeration value="summarization"/>
                  <enumeration value="abstraction"/>
                  <enumeration value="extraction"/>
                  <enumeration value="modalityTranslation"/>
                  <enumeration value="languageTranslation"/>
                  <enumeration value="colorReduction"/>
                  <enumeration value="spatialReduction"/>
                  <enumeration value="temporalReduction"/>
                  <enumeration value="samplingReduction"/>
                  <enumeration value="rateReduction"/>
                  <enumeration value="qualityReduction"/>
                  <enumeration value="compression"/>
                  <enumeration value="scaling"/>
                  <enumeration value="revision"/>
                  <enumeration value="substitution"/>
                  <enumeration value="replay"/>
                  <enumeration value="alternativeView"/>
                  <enumeration value="alternativeMediaProfile"/>
                </restriction>
              </simpleType>
              <simpleType>
                <restriction base="mpeg7:termReferenceType"/>
              </simpleType>
            </union>
          </simpleType>
        </element>
      </sequence>
      <attribute name="fidelity" type="mpeg7:zeroToOneType" use="optional"/>
      <attribute name="priority" type="nonNegativeInteger" use="optional"/>
      <attribute name="timeOffset" type="mpeg7:mediaDurationType"
        use="optional"/>
      <attribute name="timeScale" type="mpeg7:zeroToOneType" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

13.4.2.3 VariationSet DS semantics

Semantics of the VariationSetType:

Name	Definition
VariationSetType	Describes a set of variations of a multimedia content source.
Source	Describes the source multimedia content. Exactly one source shall be specified. If a source is specified within an embedded Variation, then that embedded source shall override the value of this source content for that particular Variation.
Variation	Describes the unbounded set of variations of the source multimedia content. At least one variation shall be specified.

Semantics of the VariationType:

Name	Definition
VariationType	Describes the variation multimedia content and its relationship to the source multimedia content.
Source	Describes the source multimedia content. At most one Source shall be specified. In the case that the VariationType is embedded within a VariationSet, then the specification of Source is optional.
Content	Describes the variation multimedia content.
VariationRelationship	Describes the different types of association relationships of the Variation multimedia content with respect to the source multimedia content. Multiple VariationRelationships may be specified. The types of relationships are defined as follows: • <i>summarization</i> - indicates that the variation content is a summary of the important information in the source content and allows efficient browsing and navigation. • <i>abstraction</i> - indicates that the variation content presents the salient points of the source content. • <i>extraction</i> - indicates that the variation content contains multimedia content extracted from the source content. • <i>modalityTranslation</i> - indicates that the variation content involves the conversion of the modality (image, video, text, audio, graphics) of the source content. • <i>languageTranslation</i> - indicates that the variation content involves the conversion of the language of the text or speech associated with the source content. • <i>colorReduction</i> - indicates that the variation content involves the reduction of the color detail of source visual content. • <i>spatialReduction</i> - indicates that the variation content involves the reduction in the spatial size of the source visual content. • <i>temporalReduction</i> - indicates that the variation content involves the

Name	Definition
	reduction in the time duration of the source audio or video content. • <i>samplingReduction</i> - indicates that the variation content involves the reduction in the temporal sampling period of the source audio or video content without shortening of the overall playback time. • <i>rateReduction</i> - indicates that the variation content involves the reduction in the temporal data rate of the source audio or video content. • <i>qualityReduction</i> - indicates that the variation content involves the reduction in the detail or perceived quality of the source multimedia content. • <i>compression</i> - indicates that the variation content involves the reduction in the amount of data used in representing the source multimedia content. • <i>scaling</i> - indicates that the variation content represents the source content from one of the levels in a scalable data representation. • <i>revision</i> - indicates that the variation content involves revision of the source content, such as through editing or post-processing. • <i>substitution</i> - indicates that the variation content refers to multimedia content that can generically be used as a substitute for the source content without specifying an explicit relationship between the content. • <i>replay</i> - indicates that the variation content refers to a replay such as a slow-motion replay of the source multimedia segment or content. • <i>alternativeView</i> - indicates that the variation content provides an alternative view of the source content such as a view from another camera. • <i>alternativeMediaProfile</i> - indicates that the variation content refers to an alternative profile of the source content, such as where the variation and source content were obtained from the same original multimedia content using different encoding schemes, or the variation content was obtained from the source content using transcoding. Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.
fidelity	Indicates the value of the fidelity of the variation content with respect to the source content. Fidelity is specified on a zero to one scale, where a value of one indicates the highest fidelity and zero indicates the lowest fidelity.
priority	Indicates the value of the priority of the variation content with respect to other content in a <code>VariationSet</code> . Priority is specified as a <code>nonNegativeInteger</code> , where a value of one indicates the highest priority and larger values indicate lower priorities.
timeOffset	Indicates the value of the media time offset of the variation content relative to the source content in the case that the encoding starts from a different time point of the source content. <code>TimeOffset</code> is specified using the duration of a time period. <code>TimeOffset</code> is positive if the media time point of the beginning of a given content segment in the variation content is greater than the media time point of the beginning of the same segment in the source content. Otherwise, <code>timeOffset</code> is negative.
timescale	Indicates the value of the scale factor of the media time of the variation content encoded at a different temporal rate with respect to the source content. <code>TimeScale</code> is specified on a zero to one scale where a value of one indicates the same temporal rate and 0.5 indicates that the temporal rate of the variation content is

Name	Definition
	reduced by half with respect to the source content.

13.4.2.4 VariationSet DS examples (informative)

Figure 73 illustrates a set of variations of multimedia content. The example shows the source video content in the lower left corner and shows eight variations: two variations are video content, three variations are images, two variations are text, and one variation is audio. Each variation has a fidelity value that indicates how close or faithful the variation content is to the source content.

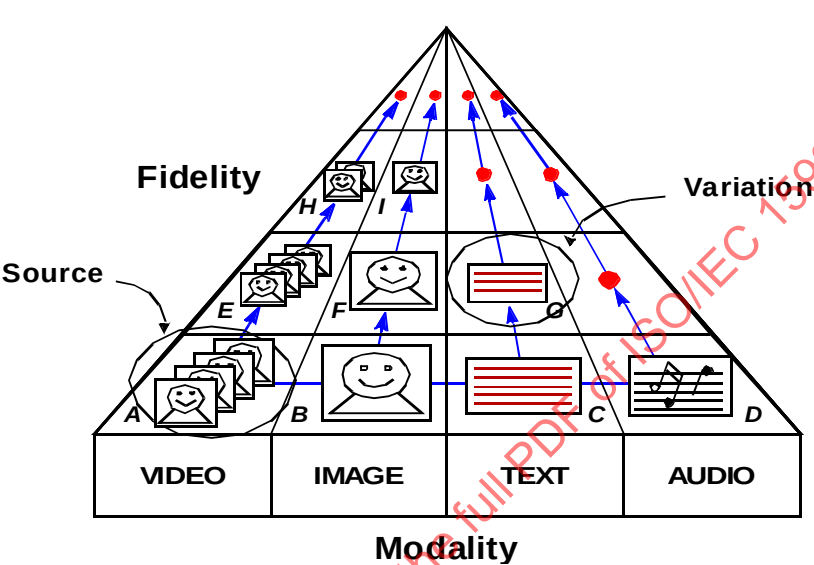


Figure 73 — Illustration of different variations of a source content. The eight variation contents have different modalities (video, image, text, audio) and fidelities with respect to the source content

The following example of the VariationSet DS describes a subset of the variation contents illustrated above, namely B, D and E. The source content is described first, using a Video element and a MediaLocator element. Following in the description is a set of three variations, described using an Image, Audio and Video element, respectively. Each Variation contains the fidelity, priority and VariationRelationship. The example shows that in some cases multiple VariationRelationships can be specified, such as the fact that variation E is derived from the source content via both SpatialReduction and Compression. Also, the timeOffset and timeScale attributes in the last Variation element relate the time-line of this variation content to the time-line of the source content.

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



```

        </Image>
    </Content>
    <VariationRelationship>extraction</VariationRelationship>
</Variation>
<Variation fidelity="0.75" priority="3">
    <Content xsi:type="AudioType">
        <Audio>
            <MediaLocator>
                <MediaUri>file://soccer-D.mp3</MediaUri>
            </MediaLocator>
        </Audio>
    </Content>
    <VariationRelationship>extraction</VariationRelationship>
    <VariationRelationship>languageTranslation</VariationRelationship>
</Variation>
<Variation fidelity="0.5" priority="2" timeOffset="PT10S" timeScale="0.5">
    <Content xsi:type="VideoType">
        <Video>
            <MediaLocator>
                <MediaUri>file://soccer-E.mpg</MediaUri>
            </MediaLocator>
        </Video>
    </Content>
    <VariationRelationship>spatialReduction</VariationRelationship>
    <VariationRelationship>compression</VariationRelationship>
    <VariationRelationship>alternativeMediaProfile</VariationRelationship>
</Variation>
</VariationSet>
</Description>
</Mpeg7>

```

14 Content organization tools

14.1 Introduction

This clause specifies tools for describing the organization and modeling of multimedia content. The tools are used for describing collections and models as follows:

Table 41 — Overview of Content Organization Tools

<i>Tools</i>	<i>Functionality</i>
Collection	Describes unordered collections. Examples include collections of multimedia content, segments, descriptors, concepts, or mixed collections.
Model	Describes parameterized models of multimedia content, descriptors, or collections (abstract).
ProbabilityModel	Describes the association of statistics or probabilities with the attributes of multimedia content, descriptors or collections.
AnalyticModel	Describes the association of labels or semantics with multimedia content or collections.
Cluster Model	Describes the association of labels or semantics, and statistics or probabilities with multimedia content collections.
Classification	Describes information about collections of multimedia content in terms of labels, semantics,

Tools	Functionality
Model	and models, which allows classification of unknown multimedia content.

14.2 Collections

This subclause specifies tools for describing collections related to multimedia content. Examples include collections of multimedia content, collections of descriptors, and collections of semantic concepts.

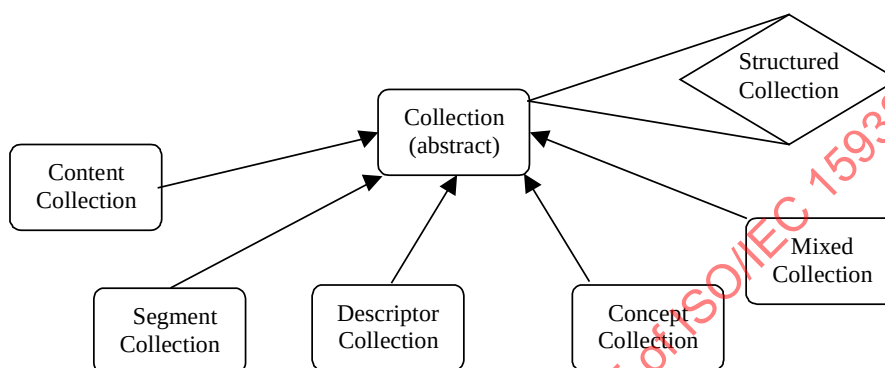


Figure 74 — Overview of the Collection tools

As shown in Figure 74, the ContentCollection DS, SegmentCollection DS, DescriptorCollection DS, ConceptCollection DS, and MixedCollection DS extend the Collection DS. The ContentCollection DS describes collections of multimedia content. The SegmentCollection DS describes collections of segments. The DescriptorCollection DS describes collections of multimedia content descriptions. The ConceptCollection DS describes collections of semantic concepts. The MixedCollection DS describes collections of content, descriptions, and concepts.

14.2.1 Collection DS

14.2.1.1 Introduction

The Collection DS describes a collection. CollectionType is an abstract type that forms the base of the different specialized types of collection description tools, including ContentCollectionType, SegmentCollectionType, DescriptorCollectionType, ConceptCollectionType, and MixedCollectionType.

14.2.1.2 Collection DS syntax

```

<!-- ##### -->
<!-- Definition of Collection DS (14.2.1) -->
<!-- ##### -->

<!-- Definition of Collection DS -->
<complexType name="CollectionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <choice minOccurs="0">
          <element name="CreationInformation"
            type="mpeg7:CreationInformationType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        <element name="CreationInformationRef" type="mpeg7:ReferenceType"/>
      </choice>
      <choice minOccurs="0">
        <element name="UsageInformation" type="mpeg7:UsageInformationType"/>
        <element name="UsageInformationRef" type="mpeg7:ReferenceType"/>
      </choice>
      <element name="TextAnnotation"
        type="mpeg7:TextAnnotationType" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="optional"/>
  </extension>
</complexContent>
</complexType>

```

14.2.1.3 Collection DS semantics

Semantics of the CollectionType:

Name	Definition
CollectionType	Describes a collection related to multimedia content (abstract). Examples include collections of multimedia content, collections of descriptors, and collections of semantic concepts. CollectionType extends DSType.
CreationInformation	Describes information related to the creation of the collection (optional).
CreationInformationRef	Describes a reference to information related to the creation of the collection (optional).
UsageInformation	Describes information related to the usage of the collection (optional).
UsageInformationRef	Describes a reference to information related to the usage of the collection (optional).
TextAnnotation	Describes a text annotation of the collection (optional).
name	Identifies the name of the collection (optional).

14.2.2 ContentCollection DS

14.2.2.1 Introduction

The ContentCollection DS describes collections of multimedia data, such as images, videos, audio, and so forth. The ContentCollection DS provides the basic functionality of describing a grouping of multimedia content into an unordered, unbounded, nested collection. ContentCollectionType extends CollectionType.

14.2.2.2 ContentCollection DS syntax

```

<!-- ##### -->
<!-- Definition of ContentCollection DS (14.2.2) -->
<!-- ##### -->

<!-- Definition of ContentCollection DS -->
<complexType name="ContentCollectionType">
  <complexContent>

```

```

<extension base="mpeg7:CollectionType">
  <sequence>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="VisualFeature" type="mpeg7:VisualDType"/>
      <element name="AudioFeature" type="mpeg7:AudioDType"/>
    </choice>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="Content" type="mpeg7:MultimediaContentType"/>
      <element name="ContentRef" type="mpeg7:ReferenceType"/>
    </choice>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="ContentCollection"
        type="mpeg7:ContentCollectionType"/>
      <element name="ContentCollectionRef" type="mpeg7:ReferenceType"/>
    </choice>
  </sequence>
</extension>
</complexContent>
</complexType>

```

14.2.2.3 ContentCollection DS semantics

Semantics of the ContentCollectionType:

Name	Definition
ContentCollectionType	Describes a collection of multimedia content, which can include images, video, audio, sounds, graphics, and so forth. The ContentCollectionType may describe a mix of different types of multimedia content within a single content collection description. ContentCollectionType extends CollectionType.
VisualFeature	Describes an aggregated visual feature of the content collection. For example, VisualFeature can describe the aggregated color of the images in an image collection using GoFGOPColor D (defined in ISO/IEC 15938-3). VisualFeature applies only in the case of a homogeneous collection of visual content such as a collection of images or a collection of video.
AudioFeature	Describes an aggregated audio feature of the content collection. The AudioFeature applies only in the case of a homogeneous collection of audio content such as a collection of songs or a collection of speech documents.
Content	Describes content that makes up the described content collection (optional).
ContentRef	Describes a reference to content that makes up the described content collection (optional).
ContentCollection	Describes a child content collection that is nested within the described content collection (optional).
ContentCollectionRef	Describes a reference to a description of a child content collection that is nested within the described content collection (optional).

14.2.2.4 ContentCollection DS examples (informative)

The following example shows the use of the ContentCollection DS to describe a collection of two images.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <Collection xsi:type="ContentCollectionType">
        <Content xsi:type="ImageType">
          <Image>
            <MediaLocator xsi:type="ImageLocatorType">
              <MediaUri>soccer1.jpg</MediaUri>
            </MediaLocator>
          </Image>
        </Content>
        <Content xsi:type="ImageType">
          <Image>
            <MediaLocator xsi:type="ImageLocatorType">
              <MediaUri>soccer2.jpg</MediaUri>
            </MediaLocator>
          </Image>
        </Content>
      </Collection>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

The following example shows the use of the ContentCollection DS to describe a collection or album of musical songs. The title of the collection is called "Live from Pisa." The collection consists of six songs, which are organized into two individual collections or album sides (side "A" and side "B"). The creation information for the collection describes the name of the artist. The creation information for each individual song describes the title of the song.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <Collection xsi:type="ContentCollectionType" name="Album - Live from Pisa">
        <CreationInformation>
          <Creation>
            <Title type="popular"> Live from Pisa </Title>
            <Creator>
              <Role href="RoleCS">
                <Name> Artist </Name>
              </Role>
              <Agent xsi:type="PersonType">
                <Name>
                  <GivenName> John </GivenName>
                  <FamilyName> Smith </FamilyName>
                </Name>
              </Agent>
            </Creator>
            <CopyrightString> Copyright 1997, All rights reserved.
            </CopyrightString>
          </Creation>
        </CreationInformation>
        <TextAnnotation>
          <FreeTextAnnotation> Collection of musical songs.
          </FreeTextAnnotation>
        </TextAnnotation>
        <ContentCollection name="Side A">
          <Content xsi:type="AudioType">
            <Audio>

```

```

        <MediaLocator>
            <MediaUri>pompodor.mid</MediaUri>
        </MediaLocator>
        <CreationInformation>
            <Creation>
                <Title> Pompodor </Title>
            </Creation>
        </CreationInformation>
    </Audio>
</Content>
<Content xsi:type="AudioType">
    <Audio>
        <MediaLocator>
            <MediaUri>wheel.mid</MediaUri>
        </MediaLocator>
        <CreationInformation>
            <Creation>
                <Title> Wheel of Misfortune</Title>
            </Creation>
        </CreationInformation>
    </Audio>
</Content>
<Content xsi:type="AudioType">
    <Audio>
        <MediaLocator>
            <MediaUri>berreta.mid</MediaUri>
        </MediaLocator>
        <CreationInformation>
            <Creation>
                <Title> Red Berreta </Title>
            </Creation>
        </CreationInformation>
    </Audio>
</Content>
</ContentCollection>
<ContentCollection name="Side B">
    <Content xsi:type="AudioType">
        <Audio>
            <MediaLocator>
                <MediaUri>brow.mid</MediaUri>
            </MediaLocator>
            <CreationInformation>
                <Creation>
                    <Title> Browbeater </Title>
                </Creation>
            </CreationInformation>
        </Audio>
    </Content>
    <Content xsi:type="AudioType">
        <Audio>
            <MediaLocator>
                <MediaUri>sunshine.mid</MediaUri>
            </MediaLocator>
            <CreationInformation>
                <Creation>
                    <Title> Sunshine </Title>
                </Creation>
            </CreationInformation>
        </Audio>
    </Content>
</ContentCollection>
<Content xsi:type="AudioType">

```

```

        <Audio>
          <MediaLocator>
            <MediaUri>corner.mid</MediaUri>
          </MediaLocator>
          <CreationInformation>
            <Creation>
              <Title> Down on the Corner </Title>
            </Creation>
          </CreationInformation>
        </Audio>
      </Content>
    </ContentCollection>
  </Collection>
</MultimediaContent>
</Description>
</Mpeg7>

```

14.2.3 SegmentCollection DS

14.2.3.1 Introduction

The SegmentCollection DS describes collections of segments, such as video segments, audio segments, still regions, and so forth, possibly from different multimedia content. The SegmentCollection DS provides the basic functionality of describing a grouping of segments into an unordered, unbounded, nested collection. SegmentCollectionType extends CollectionType.

14.2.3.2 SegmentCollection DS syntax

```

<!-- ##### -->
<!-- Definition of SegmentCollection DS (14.2.3) -->
<!-- ##### -->

<!-- Definition of SegmentCollection DS -->
<complexType name="SegmentCollectionType">
  <complexContent>
    <extension base="mpeg7:CollectionType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Segment" type="mpeg7:SegmentType"/>
          <element name="SegmentRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="SegmentCollection"
            type="mpeg7:SegmentCollectionType"/>
          <element name="SegmentCollectionRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

14.2.3.3 SegmentCollection DS semantics

Semantics of the SegmentCollectionType:

Name	Definition

Name	Definition
SegmentCollectionType	Describes a collection of segments, such as video segments, audio segments, still regions, and so forth, which are possibly from different multimedia content. The described segment collection may contain a mix of different types of segments. SegmentCollectionType extends CollectionType.
Segment	Describes a segment that makes up the described segment collection (optional).
SegmentRef	References an existing description of a segment that makes up the described segment collection (optional).
SegmentCollection	Describes a child segment collection that is nested within the described segment collection (optional).
SegmentCollectionRef	Describes a reference to a description of a child segment collection that is nested within the described segment collection (optional).

14.2.3.4 SegmentCollection DS examples (informative)

The following example illustrates the use of the SegmentCollection DS for describing a collection of two still regions. Each still region from an image corresponds to a region whose boundaries are identified using a spatial mask.

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <Collection xsi:type="SegmentCollectionType">
        <Segment xsi:type="StillRegionType">
          <MediaLocator xsi:type="ImageLocatorType">
            <MediaUri>soccer-upper-left.jpg</MediaUri>
          </MediaLocator>
          <TextAnnotation>
            <FreeTextAnnotation> Soccer image (upper left region)
            </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>
        </Segment>
        <Segment xsi:type="StillRegionType">
          <MediaLocator xsi:type="ImageLocatorType">
            <MediaUri>soccer-lower-right.jpg</MediaUri>
          </MediaLocator>
          <TextAnnotation>
            <FreeTextAnnotation> Soccer image (lower right region)
            </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>
        </Segment>
      </Collection>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

14.2.4 DescriptorCollection DS

14.2.4.1 Introduction

The DescriptorCollection DS describes collections of descriptors of multimedia content. For example, this tool can be used to describe a collection of color descriptors or shape descriptions. The DescriptorCollection DS provides the basic functionality of describing a grouping of descriptor instances into an unordered, unbounded, nested collection. DescriptorCollectionType extends CollectionType.

14.2.4.2 DescriptorCollection DS syntax

```
<!-- ##### -->
<!-- Definition of DescriptorCollection DS (14.2.4) -->
<!-- ##### -->

<!-- Definition of DescriptorCollection DS -->
<complexType name="DescriptorCollectionType">
  <complexContent>
    <extension base="mpeg7:CollectionType">
      <sequence>
        <element name="Descriptor" type="mpeg7:DType" minOccurs="0"
          maxOccurs="unbounded"/>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="DescriptorCollection"
            type="mpeg7:DescriptorCollectionType"/>
          <element name="DescriptorCollectionRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

14.2.4.3 DescriptorCollection DS semantics

Semantics of the DescriptorCollectionType:

Name	Definition
DescriptorCollectionType	Describes a collection of instances of a particular type of descriptor. The descriptor collection shall be limited to a single type of descriptor within a single description. DescriptorCollectionType extends CollectionType.
Descriptor	Describes the instance of the multimedia content descriptor by giving the value of the Descriptor (optional). The descriptor collection shall be limited to one type of Descriptor per instance.
DescriptorCollection	Describes a child descriptor collection that is nested within the described descriptor collection (optional).
DescriptorCollectionRef	Describes a reference to a description of a child descriptor collection that is nested within the described descriptor collection (optional).

14.2.4.4 DescriptorCollection DS examples (informative)

The following example shows the use of DescriptorCollection DS for describing a collection of instances of color descriptors. In this example, the color descriptors are ScalableColor D (defined in ISO/IEC 15938-3).

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <Collection xsi:type="DescriptorCollectionType">
        <Descriptor 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>
        </Descriptor>
        <Descriptor 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>
        </Descriptor>
        <Descriptor 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>
        </Descriptor>
        <Descriptor 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>
        </Descriptor>
      </Collection>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

14.2.5 ConceptCollection DS

14.2.5.1 Introduction

The ConceptCollection DS describes a collection of semantic concepts related to multimedia content, such as objects and events. The ConceptCollection DS provides the basic functionality of describing a grouping of semantic concepts into an unordered, unbounded, nested collection. ConceptCollectionType extends CollectionType.

14.2.5.2 ConceptCollection DS syntax

```
<!-- ##### -->
<!-- Definition of ConceptCollection DS (14.2.5) -->
<!-- ##### -->

<!-- Definition of ConceptCollection DS -->
<complexType name="ConceptCollectionType">
  <complexContent>
    <extension base="mpeg7:CollectionType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Concept" type="mpeg7:SemanticBaseType"/>
          <element name="ConceptRef" type="mpeg7:ReferenceType" />
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="ConceptCollection"
            type="mpeg7:ConceptCollectionType"/>
          <element name="ConceptCollectionRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

14.2.5.3 ConceptCollection DS semantics

Semantics of the ConceptCollectionType:

Name	Definition
ConceptCollectionType	Describes a collection of semantic concepts related to multimedia content, such as objects and events. The concept collection may contain a mix of different types of concepts. ConceptCollectionType extends CollectionType.
Concept	Describes a semantic concept belonging to the described concept collection (optional).
ConceptRef	Describes a reference to a semantic concept belonging to the described concept collection (optional).
ConceptCollection	Describes a child concept collection that is nested within the described concept collection (optional).
ConceptCollectionRef	Describes a reference to a description of a child concept collection that is nested within the described concept collection (optional).

14.2.5.4 ConceptCollection DS examples (informative)

The following example shows the use of the ConceptCollection DS for describing a collection of four semantic objects ("car", "house", "boat", and "tree").

```

<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <Collection xsi:type="ConceptCollectionType">
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Car </Name>
          </Label>
        </Concept>
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> House </Name>
          </Label>
        </Concept>
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Boat </Name>
          </Label>
        </Concept>
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Tree </Name>
          </Label>
        </Concept>
      </Collection>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

```

        </Collection>
    </MultimediaContent>
</Description>
</Mpeg7>

```

14.2.6 Mixed Collection DS

14.2.6.1 Introduction

The MixedCollection DS subclause specifies tools for describing mixed collections of multimedia content, descriptors and semantic concepts related to multimedia content. The MixedCollection DS provides the basic functionality of describing a grouping of these items into an unordered, unbounded, nested collection. MixedCollectionType extends CollectionType.

14.2.6.2 MixedCollection DS syntax

```

<!-- ##### -->
<!-- Definition of MixedCollection DS (14.2.6) -->
<!-- ##### -->

<!-- Definition of MixedCollection DS -->
<complexType name="MixedCollectionType">
  <complexContent>
    <extension base="mpeg7:CollectionType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Content" type="mpeg7:MultimediaContentType"/>
          <element name="ContentRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Segment" type="mpeg7:SegmentType"/>
          <element name="SegmentRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="Descriptor" type="mpeg7:DType" minOccurs="0"
          maxOccurs="unbounded" />
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Concept" type="mpeg7:SemanticBaseType"/>
          <element name="ConceptRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="MixedCollection" type="mpeg7:MixedCollectionType"/>
          <element name="MixedCollectionRef" type="mpeg7:ReferenceType"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

14.2.6.3 MixedCollection DS semantics

Semantics of the MixedCollectionType:

Name	Definition
MixedCollectionType	Describes a mixed collection of content, segments, descriptors, and semantic concepts. The MixedCollection may contain a mix of different types items within a single description. MixedCollectionType extends CollectionType.

Name	Definition
Content	Describes content included in the described mixed collection (optional).
ContentRef	Describes a reference to content included in the described mixed collection (optional).
Segment	Describes a segment included in the described mixed collection (optional).
SegmentRef	Describes a reference to a segment included in the described mixed collection (optional).
Descriptor	Describes the instance of descriptor included in the described mixed collection (optional).
Concept	Describes a concept included in the described mixed collection (optional).
ConceptRef	Describes a reference to a concept included in the described mixed collection (optional).
MixedCollection	Describes a child mixed collection that is nested within the described mixed collection (optional).
MixedCollectionRef	Describes a reference to a description of a child mixed collection that is nested within the described mixed collection (optional).

14.2.6.4 MixedCollection DS examples (informative)

The following example shows the use of the MixedCollection DS for describing a collection of two images ("baseball01.jpg" and "baseball02.jpg") and two objects ("ball" and "glove").

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <Collection xsi:type="MixedCollectionType">
        <Content xsi:type="ImageType">
          <Image>
            <MediaLocator xsi:type="ImageLocatorType">
              <MediaUri>baseball01.jpg</MediaUri>
            </MediaLocator>
          </Image>
        </Content>
        <Content xsi:type="ImageType">
          <Image>
            <MediaLocator xsi:type="ImageLocatorType">
              <MediaUri>baseball02.jpg</MediaUri>
            </MediaLocator>
          </Image>
        </Content>
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Ball </Name>
          </Label>
        </Concept>
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Glove </Name>
          </Label>
        </Concept>
      </Collection>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```

    </Collection>
  </MultimediaContent>
</Description>
</Mpeg7>

```

14.2.7 StructuredCollection DS

14.2.7.1 Introduction

The StructuredCollection DS describes the association relationships among collections (see 14.2), collection models (see 14.5.3), cluster models (see 14.6.1) and structured collections. The relationships can indicate various relations among the items, such as the similarity of collections in terms of features, or the overlapping of semantic meaning.

14.2.7.2 StructuredCollection DS syntax

```

<!-- ##### -->
<!-- Definition of StructuredCollection DS (14.2.7) -->
<!-- ##### -->

<!-- Definition of StructuredCollection DS -->
<complexType name="StructuredCollectionType" >
  <complexContent>
    <extension base="mpeg7:DSType">
      <sequence>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Collection" type="mpeg7:CollectionType"/>
          <element name="CollectionRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="CollectionModel" type="mpeg7:CollectionModelType"/>
          <element name="CollectionModelRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="ClusterModel" type="mpeg7:ClusterModelType"/>
          <element name="ClusterModelRef" type="mpeg7:ReferenceType"/>
        </choice>
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="StructuredCollection"
            type="mpeg7:StructuredCollectionType"/>
          <element name="StructuredCollectionRef" type="mpeg7:ReferenceType"/>
        </choice>
        <element name="Relationships" type="mpeg7:GraphType" minOccurs="1"
          maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

14.2.7.3 StructuredCollection DS semantics

Semantics of the StructuredCollectionType:

Name	Definition
StructuredCollectionType	Describes the association relationships among collections, collection models, cluster models, and structured collections.
Collection	Describes an unbounded set of collections (optional).
CollectionRef	Describes an unbounded set of references to collections (optional).
CollectionModel	Describes an unbounded set of collection models (optional).
CollectionModelRef	Describes an unbounded set of references to collection models (optional).
ClusterModel	Describes an unbounded set of cluster models (optional).
ClusterModelRef	Describes an unbounded set of references to cluster models (optional).
StructuredCollection	Describes an unbounded set of structured collections (optional).
StructuredCollectionRef	Describes an unbounded set of references to structured collections (optional).
Relationships	Describes a graph that gives an unbounded set of relations among the collections, collection models and cluster models.

14.2.7.4 StructuredCollection DS examples (informative)

The following example shows the use of the StructuredCollection DS for describing a structured collection consisting of two collections of images, each containing two images. The following relationships are described between the two collections of images using a relationship graph: similarTo and overlaps, which gives that the two collections are similar, and they overlap in some sense.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="MultimediaCollectionType">
      <StructuredCollection>
        <!-- Describes a Collection of two images -->
        <Collection id="collection1" xsi:type="ContentCollectionType">
          <Content xsi:type="ImageType">
            <Image>
              <MediaLocator xsi:type="ImageLocatorType">
                <MediaUri>soccer1.jpg</MediaUri>
              </MediaLocator>
            </Image>
          </Content>
          <Content xsi:type="ImageType">
            <Image>
              <MediaLocator xsi:type="ImageLocatorType">
                <MediaUri>soccer2.jpg</MediaUri>
              </MediaLocator>
            </Image>
          </Content>
        </Collection>
        <!-- Describes a Collection of two images -->
```

```

<Collection id="collection2" xsi:type="ContentCollectionType">
  <Content xsi:type="ImageType">
    <Image>
      <MediaLocator xsi:type="ImageLocatorType">
        <MediaUri>soccer3.jpg</MediaUri>
      </MediaLocator>
    </Image>
  </Content>
  <Content xsi:type="ImageType">
    <Image>
      <MediaLocator xsi:type="ImageLocatorType">
        <MediaUri>soccer4.jpg</MediaUri>
      </MediaLocator>
    </Image>
  </Content>
</Collection>
<!-- Describes the relationships among the collections -->
<Relationships>
  <Node id="source" href="#collection1"/>
  <Node id="target1" href="#collection2"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:similar"
    source="#source" target="#target1"/>
  <Relation
    type="urn:mpeg:mpeg7:cs:BaseRelationCS:2001:overlaps"
    source="#source" target="#target1"/>
</Relationships>
</StructuredCollection>
</MultimediaContent>
</Description>
</Mpeg7>

```

14.3 Models

14.3.1 Introduction

This subclause specifies tools for describing models of multimedia content. The models provide parameterized descriptions of multimedia content or collections of multimedia content. In the case of cluster models, the models are described in terms of statistics or probabilities associated with the attributes of collections of multimedia content. In the case of collection models, the models are described through examples of multimedia content.

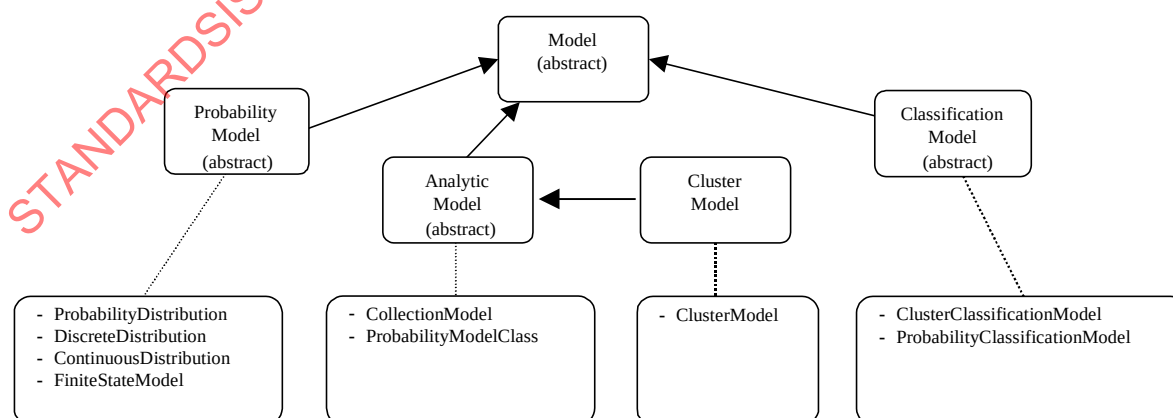


Figure 75 — Overview of tools for describing models of multimedia content

14.3.2 Model DS

14.3.2.1 Introduction

The Model DS characterizes the attributes and features of multimedia content. ModelType is an abstract type that forms the base of the different specialized model types, that characterize multimedia content on the basis of probabilities, statistics, and examples.

14.3.2.2 Model DS syntax

```
<!-- ##### -->
<!-- Definition of Model DS (14.3.2) -->
<!-- ##### -->

<!-- Definition of Model DS -->
<complexType name="ModelType" abstract="true">
  <complexContent>
    <extension base="mpeg7:DSType">
      <attribute name="confidence" type="mpeg7:zeroToOneType" use="optional"/>
      <attribute name="reliability" type="mpeg7:zeroToOneType" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

14.3.2.3 Model DS semantics

Semantics of the ModelType:

Name	Definition
ModelType	Describes a model of multimedia content (abstract). ModelType extends DSType.
confidence	Identifies the confidence of the Model (optional). Confidence indicates the degree of confidence in the model. For example, in the case of a probability model, if the model parameter values are far away from their corresponding statistical mean or outside a corresponding confidence interval, the model shall be assigned a lower confidence score. Likewise if the model parameters values are considered to be unusual, then the confidence in the values may be low. Otherwise, if the values are usual or within an expected range, then the confidence may be high. A value of 1.0 indicates highest confidence. A value of 0.0 indicates lowest confidence.
reliability	Identifies the reliability of the model (optional). Reliability refers to the accuracy of the values of the model parameters. Reliability may take into account the method of extraction. For example, in some cases if model parameter values are extracted automatically, then a reliability score may be assigned that is lower than those extracted manually. A value of 1.0 indicates highest reliability. A value of 0.0 indicates lowest reliability.

14.4 Probability models

14.4.1 Introduction

This subclause specifies tools for describing models of multimedia content using probabilities and statistics.

14.4.2 ProbabilityModel DS

14.4.2.1 Introduction

ProbabilityModel DS describes the statistical modelling of multimedia content by describing the attributes and features of multimedia content using probabilities and statistics. ProbabilityModelType is an abstract type that forms the base type of the different specialized probability model types.

14.4.2.2 ProbabilityModel DS syntax

```
<!-- ##### -->
<!-- Definition of ProbabilityModel DS (14.4.2) -->
<!-- ##### -->

<!-- Definition of ProbabilityModel DS -->
<complexType name="ProbabilityModelType" abstract="true">
  <complexContent>
    <extension base="mpeg7:ModelType"/>
  </complexContent>
</complexType>
```

14.4.2.3 ProbabilityModel DS semantics

Semantics of the ProbabilityModelType:

Name	Definition
ProbabilityModelType	Describes the statistical modelling of multimedia content by describing the attributes and features of multimedia content using probabilities and statistics. (abstract). ProbabilityModelType extends ModelType.

14.4.3 ProbabilityDistribution DS

14.4.3.1 Introduction

Probability distributions provide a basic tool for building statistical models of multimedia content. In general, probability distributions describe the statistical properties of random variables in the form of a distribution. For many purposes, it is not necessary to represent all information about a distribution, rather, pieces of information may suffice for describing a distribution. For example, the information about the location (mean) and dispersion (variance) is most frequently used.

This subclause provides tools for describing both discrete and continuous probability distributions. In the discrete case, different classes of parameterized probability mass functions are described through the specification of the various parameters (i.e., probability of success, number of trials, and so forth). Furthermore, arbitrary discrete probability mass functions can be described by specifying the values of the distribution directly using histograms. In the continuous case, the probability density functions are described through the specification of various parameters (i.e., mean, variance, moments, shape, and so forth).

The ProbabilityDistributionType is the base type of the discrete and continuous probability distributions. The ProbabilityDistributionType includes attributes that are common to all distributions such as mean, variance, min, max, mode, median, moment, and so forth. In general, the probability distributions can have multiple dimensions, therefore, the ProbabilityDistributionType describes the parameters of the distributions as vectors. In the case that a distribution is one-dimensional, the first element of each vector shall be used to specify the corresponding parameter as a scalar quantity.

14.4.3.2 ProbabilityDistribution DS syntax

```

<!-- ##### -->
<!-- Definition of ProbabilityDistribution DS (14.4.3) -->
<!-- ##### -->

<!-- Definition of ProbabilityDistribution DS -->
<complexType name="ProbabilityDistributionType">
  <complexContent>
    <extension base="mpeg7:ProbabilityModelType">
      <sequence>
        <element name="Mean" type="mpeg7:DoubleMatrixType" minOccurs="0"/>
        <element name="Variance" type="mpeg7:DoubleMatrixType"
          minOccurs="0"/>
        <element name="Min" type="mpeg7:DoubleMatrixType" minOccurs="0"/>
        <element name="Max" type="mpeg7:DoubleMatrixType" minOccurs="0"/>
        <element name="Mode" type="mpeg7:DoubleMatrixType" minOccurs="0"/>
        <element name="Median" type="mpeg7:DoubleMatrixType"
          minOccurs="0"/>
        <element name="Moment" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="Value">
                <complexType>
                  <simpleContent>
                    <extension base="mpeg7:DoubleMatrixType">
                      <attribute name="order"
                        type="nonNegativeInteger"
                        use="required"/>
                    </extension>
                  </simpleContent>
                </complexType>
              </element>
              <element name="Center" type="mpeg7:DoubleMatrixType"
                minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
        <element name="Cumulant" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="Value">
                <complexType>
                  <simpleContent>
                    <extension base="mpeg7:DoubleMatrixType">
                      <attribute name="order"
                        type="nonNegativeInteger"
                        use="required"/>
                    </extension>
                  </simpleContent>
                </complexType>
              </element>
              <element name="Center" type="mpeg7:DoubleMatrixType"
                minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
      <attribute name="dim" type="nonNegativeInteger" default="1"/>
    </extension>
  </complexContent>
</complexType>

```

```

</extension>
</complexContent>
</complexType>

```

14.4.3.3 ProbabilityDistribution DS semantics

Semantics of the ProbabilityDistributionType:

Name	Definition
ProbabilityDistributionType	Describes a probability distribution. ProbabilityDistributionType extends ProbabilityModelType.
Mean	Describes the mean of the probability distribution (optional). The Mean is specified as a vector in order to allow description of the mean of multi-dimensional probability distributions. Given a one-dimensional distribution, the Mean takes a scalar value and is given by the first element of the vector. In general, the Mean of the distribution is given by the mean vector: $Mean = (\mu_{x_1}, \mu_{x_2}, \Lambda, \mu_{x_k})$, which locates the probability distribution in the k dimensional space of points (x_1, x_2, Λ, x_k), where μ_{x_i} is computed as follows for the case of a discrete multi-dimensional distribution $p(x_1, x_2, \Lambda, x_k)$: $\mu_{x_i} = E[X_i] = \sum_{x_1} \sum_{x_2} \Lambda \sum_{x_k} x_i p(x_1, x_2, \Lambda, x_k),$ and as follows: $\mu_{x_i} = E[X_i] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Lambda \int_{-\infty}^{\infty} x_i f(x_1, x_2, \Lambda, x_k) dx_1 dx_2 \Lambda dx_k$ for the case of a continuous multi-dimensional distribution $f(x_1, x_2, \Lambda, x_k)$.
Variance	Describes the variance of the probability distribution (optional). The Variance is specified as a vector in order to allow description of the variance of multi-dimensional probability distributions. In general, the Variance of a multi-dimensional distribution is given by the variance vector: $Variance = (\sigma_{x_1}^2, \sigma_{x_2}^2, \Lambda, \sigma_{x_k}^2)$, where the individual variances $\sigma_{x_i}^2$ serve as measures of the dispersion of probability in the x_i direction in the k dimensional space, where $i = 1, 2, K, k$. The variance $\sigma_{x_i}^2$ along the x_i direction is computed as follows for the case of a discrete multi-dimensional distribution $p(x_1, x_2, \Lambda, x_k)$: $\sigma_{x_i}^2 = E[(X_i - \mu_{x_i})^2] = \sum_{x_1} \sum_{x_2} \Lambda \sum_{x_k} (x_i - \mu_{x_i})^2 p(x_1, x_2, \Lambda, x_k),$ and as follows:

Name	Definition
	$\sigma_{X_i}^2 = E[(X_i - \mu_{X_i})^2] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Lambda \int_{-\infty}^{\infty} (x_i - \mu_{X_i})^2 f(x_1, x_2, \Lambda, x_k) dx_1 dx_2 \Lambda dx_k$ for the case of a continuous multi-dimensional distribution $f(x^1, x^2, \Lambda, x_k)$.
Min	Describes the minimum value of the probability distribution (optional). The Min is specified as a vector in order to allow description of the minimum of multi-dimensional probability distributions. The Min of the multi-dimensional distribution is computed by individually computing the minimum along each dimension as follows: $Min = (\min(p_{X_1}(x_1)), \min(p_{X_2}(x_2)), \Lambda, \min(p_{X_k}(x_k)))$, where $p_{X_i}(x_i)$ gives the marginal function $p_{X_i}(x_i) = \sum_{\forall x_j \neq x_i} p(x_1, x_2, \Lambda, x_k)$, and $\min(p_{X_i}(x_i))$ gives the minimum value of the distribution range along the x_i direction of the distribution in the k dimensional space, where $i = 1, 2, K, k$.
Max	Describes the maximum value of the probability distribution (optional). The Max is specified as a vector in order to allow description of the maximum of multi-dimensional probability distributions. The Max of the multi-dimensional distribution is computed by individually computing the maximum along each dimension as follows: $Max = (\max(p_{X_1}(x_1)), \max(p_{X_2}(x_2)), \Lambda, \max(p_{X_k}(x_k)))$, where $p_{X_i}(x_i)$ gives the marginal function $p_{X_i}(x_i) = \sum_{\forall x_j \neq x_i} p(x_1, x_2, \Lambda, x_k)$, and $\max(p_{X_i}(x_i))$ gives the maximum value of the distribution range along the x_i direction of the distribution in the k dimensional space, where $i = 1, 2, K, k$.
Mode	Describes the mode of the probability distribution (optional). The Mode is specified as a vector in order to allow description of the mode of multi-dimensional probability distributions. The Mode of the multi-dimensional distribution is computed by individually computing the mode along each dimension as follows: $Mode = (\text{mode}(p_{X_1}(x_1)), \text{mode}(p_{X_2}(x_2)), \Lambda, \text{mode}(p_{X_k}(x_k)))$, where $p_{X_i}(x_i)$ gives the marginal function $p_{X_i}(x_i) = \sum_{\forall x_j \neq x_i} p(x_1, x_2, \Lambda, x_k)$, and $\text{mode}(p_{X_i}(x_i))$ gives the mode, or most probable value, of the distribution along the x_i direction in the k dimensional space, where $i = 1, 2, K, k$. For a continuous distribution, the mode corresponds to the value at which the probability density function is maximum, that is, $\text{mode} = x_m$, where $\forall_j p_{X_j}(x_j) \leq p_{X_m}(x_m)$. The mode need not be uniquely defined since there can be more than one most probable value of a distribution.

Name	Definition
Median	Describes the median of the probability distribution (optional). The Median is specified as a vector in order to allow description of the media of multi-dimensional probability distributions. The Median of the multi-dimensional distribution is computed by individually computing the median along each dimension as follows: where $p_{X_i}(x_i)$ gives the marginal function $p_{X_i}(x_i) = \sum_{\forall x_j \neq x_i} p(x_1, x_2, \Lambda, x_k)$, and $med(p_{X_i}(x_i))$ gives the median of the distribution along the x_i direction in the k dimensional space, where $i=1,2,K,k$. The value of the median along the x_i direction can be obtained analytically by solving the equation $F_{X_i}(x_i)=1/2$, where $F_{X_i}(x_i)$ gives the marginal cumulative density function along the x_i direction, where $F_{X_i}(x_i) = P\{X_i \leq x_i\}$. The median provides one measure of the location for the distribution.
Moment	Describes the value of the r^{th} central moment of the probability distribution around a point c, where r is the order of the moment and c is the value of the point around which the moment is computed (optional). The Moment is specified as a vector in order to allow description of multi-dimensional probability distributions. In general, the r^{th} moment of a multi-dimensional distribution around a center c is given by the vector: $Moment(r, c) = (\mu_{X_1}^{(r,c)}, \mu_{X_2}^{(r,c)}, \Lambda, \mu_{X_k}^{(r,c)})$. The moment $\mu_{X_i}^{(r,c)}$ along the x_i direction is computed as follows for the case of a discrete multi-dimensional distribution $p(x_1, x_2, \Lambda, x_k)$: $\mu_{X_i}^{(r,c)} = E[(X_i - c_i)^r] = \sum_{x_1} \sum_{x_2} \Lambda \sum_{x_k} (x_i - c_i)^r p(x_1, x_2, \Lambda, x_k),$ and as follows: $\mu_{X_i}^{(r,c)} = E[(X_i - c_i)^r] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Lambda \int_{-\infty}^{\infty} (x_i - c_i)^r f(x_1, x_2, \Lambda, x_k) dx_1 dx_2 \Lambda dx_k$ for the case of a continuous multi-dimensional distribution $f(x_1, x_2, \Lambda, x_k)$.
Value	Describes the value of the moment of the probability distribution around a point specified by it Center.
order	Specifies the order of the moment. For example order="1" corresponds to the mean; order="2" corresponds to the variance; order="3" corresponds to the skewness; order="4" corresponds to kurtosis, and so forth. In the specification of moment above, the order of the cumulant is denoted by the symbol r .

Name	Definition
Center	Specifies the point around which the moment is computed (optional). If not specified, the Center shall have the value of the mean of the distribution. In this case, the moments are considered to be central moments.
Cumulant	Describes the value of the s^{th} cumulant of the probability distribution, where s is the order of the cumulant. The Cumulant is specified as a vector to allow description of a multi-dimensional probability distribution. The cumulant generating function of a random variable X, whether discrete or continuous, is defined by $K_X(\Theta) = \log M_X(\Theta)$, where $M_X(\Theta) = E(e^{\Theta X})$ gives the moment generating function, which is defined for all values Θ for which the expect value exists, where Θ gives the center point around which the cumulant is computed. Thus, the moment generating function $M_X(\Theta)$ along the x_i direction is computed as follows for the case of a discrete multi-dimensional distribution $p(x_1, x_2, \Lambda, x_k)$: $M_{X_i}(\Theta) = E(e^{\Theta X_i}) = \sum_{x_1} \sum_{x_2} \Lambda \sum_{x_k} e^{\Theta X_i} p(x_1, x_2, \Lambda, x_k),$ and as follows: $M_{X_i}(\Theta) = E(e^{\Theta X_i}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Lambda \int_{-\infty}^{\infty} e^{\Theta X_i} f(x_1, x_2, \Lambda, x_k) dx_1 dx_2 \Lambda dx_k$ for the case of a continuous multi-dimensional distribution $f(x_1, x_2, \Lambda, x_k)$. Letting $K_X^{[s]}(\Theta) = (d/d\Theta)^s K_X(\Theta)$, the quantities $\kappa_s = K_X^{[s]}(0)$, $s=1,2,\dots$, are known as the s^{th} cumulants of the random variable X. In particular, $\kappa_1 = \mu_X$ (mean) and $\kappa_2 = \sigma_X^2$ (variance).
Value	Describes the value of the cumulant of the probability distribution around a point specified by Center.
order	Specifies the order of the cumulant. For example, given a zero Center, for order = 1, the cumulant is equivalent to the mean. Given a zero Center, for order = 2, the cumulant is equivalent to the variance. In the specification of cumulant above, the order of the cumulant is denoted by the symbol s .
Center	Specifies the point around which the cumulant is computed (optional). If not specified, the Center shall have the value of the zero vector.
dim	Identifies the dimensionality of the probability distribution (optional). The default value of dim is one. Given the value of dim="1", then the mean, variance, min, max, mode, median, moment and cumulant are scalar quantities.

14.4.3.4 ProbabilityDistribution DS examples (informative)

The following example shows the use of the ProbabilityDistribution DS for describing a probability distribution in terms of its statistics such as mean, variance, min, max, mode, median, and moment.

```
<Mpeg7>
  <DescriptionUnit xsi:type="ProbabilityDistributionType" confidence="1.0" dim="2">
    <Mean mpeg7:dim="2"> 0.25 0.5 </Mean>
    <Variance mpeg7:dim="2"> 0.1 0.9 </Variance>
    <Min mpeg7:dim="2"> 0.0 0.0 </Min>
    <Max mpeg7:dim="2"> 1.0 1.0 </Max>
    <Mode mpeg7:dim="2"> 0.5 0.5 </Mode>
    <Median mpeg7:dim="2"> 0.5 0.5 </Median>
    <Moment>
      <Value mpeg7:dim="2" order="3"> 0.2 0.3 </Value>
    </Moment>
  </DescriptionUnit>
</Mpeg7>
```

14.4.4 Discrete distribution description tools

14.4.4.1 Introduction

This subclause specifies tools for describing discrete probability distributions including discrete probability distributions and parameterized discrete probability mass functions. The parameterized discrete probability mass functions are described using the parameters of the distribution functions (i.e., probability of success, number of trials, and so forth). In the case of a one-dimensional discrete probability mass function, the discrete probability distributions is described by the values of the distribution directly using a histogram. The parameterized discrete distributions can be multi-dimensional, in which case the parameters are described in terms of multi-dimensional vectors.

There are advantages of using parameterized distributions compared to describing the values of the distributions directly. First, the parameterized distributions can describe multi-dimensional distributions, and second, the description of the discrete distributions using parameterized functions is more compact. For example, consider the description of a uniform one-dimensional discrete distribution that has 256 values. Representing the distribution using a histogram requires a vector of size 256. Representing the distribution using a parameterized uniform distribution requires only a single parameter that indicates the value of the distribution, or at most three parameters in the case that the distribution is scaled and shifted. However, in many cases the parameterised representation provides only an approximation of the actual distribution.

DiscreteDistributionType is an abstract type that forms the base type of the different specialized types of discrete probability distributions.

14.4.4.2 Discrete distribution description tools syntax

```
<!-- ##### -->
<!-- Definition of Discrete distribution DSs (14.4.4) -->
<!-- ##### -->

<!-- Definition of DiscreteDistribution DS -->
<complexType name="DiscreteDistributionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:ProbabilityDistributionType"/>
  </complexContent>
</complexType>

<!-- Definition of HistogramProbability DS -->
<complexType name="HistogramProbabilityType">
```

```

<complexContent>
  <extension base="mpeg7:DiscreteDistributionType">
    <sequence>
      <element name="Distribution" type="mpeg7:ProbabilityMatrixType"
        minOccurs="0"/>
    </sequence>
    <attribute name="momentNormalized" type="mpeg7:nonNegativeReal"
      use="optional"/>
    <attribute name="count" type="nonNegativeInteger" use="optional"/>
  </extension>
</complexContent>
</complexType>

<!-- Definition of BinomialDistribution DS -->
<complexType name="BinomialDistributionType">
  <complexContent>
    <extension base="mpeg7:DiscreteDistributionType">
      <sequence>
        <element name="SuccessProbability"
          type="mpeg7:ProbabilityMatrixType"/>
        <element name="NumOfTrials" type="mpeg7:IntegerMatrixType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of HyperGeometricDistribution DS -->
<complexType name="HyperGeometricDistributionType">
  <complexContent>
    <extension base="mpeg7:DiscreteDistributionType">
      <sequence>
        <element name="NumOfSuccesses" type="mpeg7:IntegerMatrixType"/>
        <element name="NumOfTrials" type="mpeg7:IntegerMatrixType"/>
        <element name="PopulationSize" type="mpeg7:IntegerMatrixType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of PoissonDistribution DS -->
<complexType name="PoissonDistributionType">
  <complexContent>
    <extension base="mpeg7:DiscreteDistributionType">
      <sequence>
        <element name="Lambda" type="mpeg7:DoubleMatrixType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of GeometricDistribution DS -->
<complexType name="GeometricDistributionType">
  <complexContent>
    <extension base="mpeg7:DiscreteDistributionType">
      <sequence>
        <element name="SuccessProbability" type="mpeg7:DoubleMatrixType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```



```

<!-- Definition of DiscreteUniformDistribution DS -->
<complexType name="DiscreteUniformDistributionType">
  <complexContent>
    <extension base="mpeg7:DiscreteDistributionType">
      <sequence>
        <element name="Range" type="mpeg7:DoubleMatrixType"/>
        <element name="Scale" type="mpeg7:DoubleMatrixType" minOccurs="0"/>
        <element name="Shift" type="mpeg7:DoubleMatrixType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

14.4.4.3 Discrete distribution description tools semantics

Semantics of the DiscreteDistributionType:

Name	Definition
DiscreteDistributionType	Describes a discrete probability distribution (abstract). DiscreteDistributionType extends ProbabilityDistributionType.

Semantics of the HistogramProbabilityType:

Name	Definition
HistogramProbabilityType	Describes a one-dimensional discrete probability mass function by giving the distribution values directly using a histogram: $p(x)$. HistogramProbabilityType extends DiscreteDistributionType.
Distribution	Describes the values of the one-dimensional discrete probability mass function. The values of the distribution fall in the range 0.0 to 1.0 as follows: $0.0 \leq p(x) \leq 1.0$.
momentNormalized	Identifies the normalization of the discrete probability mass function (optional). A value of momentNormalized=1 indicates that the elements of the distribution sum to one as follows: $\sum_x p(x) = 1$. A value of momentNormalized=2 indicates that the squares of the elements sum to one as follows: $\sum_x p(x)^2 = 1$. Other values of momentNormalized= n indicates that the n^{th} power of the elements sum to one as follows: $\sum_x p(x)^n = 1$.
count	Specifies the number of non-empty elements or non-zero values in the histogram (optional), i.e., where $p(x) \neq 0.0$.

Semantics of the BinomialDistributionType:

Name	Definition
BinomialDistributionType	Describes a discrete multi-dimensional probability mass function that takes the form of a binomial distribution by specifying the parameters of the distribution. The probability mass function for the binomial distribution along the dimension z is given as follows: $p_z(k) = \binom{n}{k} p^k (1-p)^{n-k}$, for $k = 0, 1, \dots, n$ where the parameter p specifies the success probability and the parameter n specifies the number of trials. BinomialDistributionType extends DiscreteDistributionType.
SuccessProbability	Describes the parameter p needed to represent the binomial distribution. SuccessProbability corresponds to the probability of success in each of the Bernoulli trials.
NumOfTrials	Describes the parameter n needed to represent the binomial distribution. NumOfTrials specifies the number of Bernoulli trials.

Semantics of the HyperGeometricDistributionType:

Name	Definition
HyperGeometricDistributionType	Describes a discrete multi-dimensional probability mass function that takes the form of a hypergeometric distribution by specifying the parameters of the distribution. The probability mass function for the HyperGeometric Distribution is given as follows: $p_z(k) = \frac{\binom{s}{k} \binom{N-s}{n-k}}{\binom{N}{n}}$, for $k = 0, 1, \dots, n$, where the random variable "k" is the exact number of successes that occur in "n" trials, "N" is the size of the population, and "s" is the number of successes in the sample. HyperGeometricDistributionType extends DiscreteDistributionType.
NumOfSuccesses	Describes the parameter s needed to represent the HyperGeometric Distribution. NumOfSuccesses specifies the number of successes in the sample.
NumOfTrials	Describes the parameter n needed to represent the HyperGeometric Distribution. NumOfTrials specifies the number of Bernoulli trials.
PopulationSize	Describes the parameter N , where $N = 1, 2, \dots$, which is needed to represent the hypergeometric distribution. PopulationSize specifies the size of the population of sample.

Semantics of the PoissonDistributionType:

Name	Definition
PoissonDistributionType	Describes a discrete multi-dimensional probability mass function that takes the form of a Poisson distribution by specifying the parameters of the distribution. The probability mass function for the Poisson distribution is given as follows: $p_z(k) = \frac{e^{-\lambda} \lambda^k}{k!}, \text{ for } k = 0, 1, 2, \dots \text{ where } \lambda > 0 \text{ specifies the parameter of the discrete distribution. PoissonDistributionType extends DiscreteDistributionType.}$
Lambda	Describes the parameter lambda ($\lambda > 0$) needed to represent the Poisson distribution. Lambda is defined as a vector in order to allow the description of multi-dimensional distributions.

Semantics of the GeometricDistributionType:

Name	Definition
GeometricDistributionType	Describes a discrete multi-dimensional probability mass function that takes the form of a geometric distribution by specifying the parameters of the distribution. The probability mass function for the geometric distribution is given as follows: $p_z(k) = q^{k-1} p$, for $k = 1, 2, \dots$ where the parameter p specifies the success probability and $q = 1 - p$. GeometricDistributionType extends DiscreteDistributionType.
SuccessProbability	Describes the parameter p needed to represent the geometric distribution. SuccessProbability corresponds to the probability of success in each of the Bernoulli trials.

Semantics of the DiscreteUniformDistributionType:

Name	Definition
DiscreteUniformDistributionType	Describes a discrete multi-dimensional probability mass function that takes the form of a uniform distribution by specifying the parameters of the distribution. The probability mass function for the discrete uniform distribution is given as follows: $p_z(k) = \frac{1}{K}$, for $k = c + h, c + 2h, \dots, c + Kh$ where the parameter K specifies value of the distribution and h specifies the scale factor (default=1), and c specifies the shift factor (default=0). DiscreteUniformDistributionType extends DiscreteDistributionType.
Range	Describes the range parameter K needed to represent the uniform distribution. The value corresponds to the value of each element of the distribution, which is flat.
Scale	Describes the scaling factor h of the uniform distribution (optional)

Name	Definition
Shift	Describes the shifting factor c of the uniform distribution (optional).

14.4.4.4 Discrete distribution description tools examples (informative)

The following example illustrates the use of the HistogramProbability DS for describing an eight-dimensional histogram.

```
<Mpeg7>
  <DescriptionUnit xsi:type="HistogramProbabilityType" momentNormalized="1"
    dim="8">
    <Distribution mpeg7:dim="8"> 0.125 0.0 0.0 0.25 0.125 0.25 0.25 0.0
    </Distribution>
  </DescriptionUnit>
</Mpeg7>
```

The following example illustrates the use of the BinomialDistribution DS for describing a one-dimensional binomial distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="BinomialDistributionType" dim="1">
    <SuccessProbability mpeg7:dim="1"> 0.5 </SuccessProbability>
    <NumOfTrials mpeg7:dim="1"> 16 </NumOfTrials>
  </DescriptionUnit>
</Mpeg7>
```

The following example illustrates the use of the BinomialDistribution DS for describing a two-dimensional hypergeometric distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="HyperGeometricDistributionType" dim="2">
    <NumOfSuccesses mpeg7:dim="2"> 5 5 </NumOfSuccesses>
    <NumOfTrials mpeg7:dim="2"> 8 8 </NumOfTrials>
    <PopulationSize mpeg7:dim="2"> 16 16 </PopulationSize>
  </DescriptionUnit>
</Mpeg7>
```

The following example illustrates the use of the PoissonDistribution DS for describing a three-dimensional Poisson distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="PoissonDistributionType" dim="3">
    <Lambda mpeg7:dim="3"> 0.4 0.3 0.75 </Lambda>
  </DescriptionUnit>
</Mpeg7>
```

The following example illustrates the use of the GeometricDistribution DS for describing a one-dimensional geometric distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="GeometricDistributionType" dim="1">
```

```

    <SuccessProbability mpeg7:dim="1"> 0.75 </SuccessProbability>
  </DescriptionUnit>
</Mpeg7>

```

The following example illustrates the use of the DiscreteUniformDistribution DS for describing a one-dimensional discrete uniform distribution.

```

<Mpeg7>
  <DescriptionUnit xsi:type="DiscreteUniformDistributionType" dim="1">
    <Range mpeg7:dim="1"> 0.125 </Range>
    <Scale mpeg7:dim="1"> 2.0 </Scale>
    <Shift mpeg7:dim="1"> 8.0 </Shift>
  </DescriptionUnit>
</Mpeg7>

```

14.4.5 Continuous distribution description tools

14.4.5.1 Introduction

This subclause specifies tools for describing continuous probability distributions. The probability density functions are described through the specification of various parameters (i.e., mean, variance, moments, shape, and so forth) of known distribution functions. The advantage of using parameterized continuous distributions is that known probability density functions can be specified in a compact form requiring only specification of parameters of the distribution. Furthermore, in many cases, the computation of mean and variance can be made directly from the parameters using known formulas.

ContinuousDistributionType is an abstract type that forms the base type of the different specialized types of continuous probability distributions.

14.4.5.2 Continuous distribution description tools syntax

```

<!-- ##### -->
<!-- Definition of Continuous distribution DSs (14.4.5) -->
<!-- ##### -->

<!-- Definition of ContinuousDistribution DS -->
<complexType name="ContinuousDistributionType" abstract="true">
  <complexContent>
    <extension base="mpeg7:ProbabilityDistributionType"/>
  </complexContent>
</complexType>

<!-- Definition of GaussianDistribution DS -->
<complexType name="GaussianDistributionType">
  <complexContent>
    <extension base="mpeg7:ContinuousDistributionType">
      <sequence>
        <element name="Determinant" type="double" minOccurs="0"/>
        <element name="CovarianceInverse" type="mpeg7:DoubleMatrixType"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of GeneralizedGaussianDistribution DS -->
<complexType name="GeneralizedGaussianDistributionType">

```

```

    <complexContent>
      <extension base="mpeg7:GaussianDistributionType">
        <sequence>
          <element name="Shape" type="mpeg7:DoubleMatrixType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

<!-- Definition of ExponentialDistribution DS -->
<complexType name="ExponentialDistributionType">
  <complexContent>
    <extension base="mpeg7:ContinuousDistributionType">
      <sequence>
        <element name="Theta" type="mpeg7:DoubleMatrixType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of GammaDistribution DS -->
<complexType name="GammaDistributionType">
  <complexContent>
    <extension base="mpeg7:ContinuousDistributionType">
      <sequence>
        <element name="Theta" type="mpeg7:DoubleMatrixType"/>
        <element name="Shape" type="mpeg7:DoubleMatrixType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of ContinuousUniformDistribution DS -->
<complexType name="ContinuousUniformDistributionType">
  <complexContent>
    <extension base="mpeg7:ContinuousDistributionType">
      <sequence>
        <element name="MaxRange" type="mpeg7:DoubleMatrixType"/>
        <element name="MinRange" type="mpeg7:DoubleMatrixType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of LognormalDistribution DS -->
<complexType name="LognormalDistributionType">
  <complexContent>
    <extension base="mpeg7:ContinuousDistributionType">
      <sequence>
        <element name="NormalMean" type="mpeg7:DoubleMatrixType"/>
        <element name="NormalVariance" type="mpeg7:DoubleMatrixType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of GaussianMixtureModel DS -->
<complexType name="GaussianMixtureModelType">
  <complexContent>
    <extension base="mpeg7:ContinuousDistributionType">

```

```

<sequence maxOccurs="1" maxOccurs="unbounded">
  <element name="Weight" type="mpeg7:nonNegativeReal"/>
  <element name="GaussianDistribution"
    type="mpeg7:GaussianDistributionType"/>
</sequence>
</extension>
</complexContent>
</complexType>

```

14.4.5.3 Continuous distribution description tools semantics

Semantics of the ContinuousDistributionType:

Name	Definition
ContinuousDistributionType	Describes a continuous multi-dimensional probability density function (abstract). ContinuousDistributionType extends ProbabilityDistributionType.

Semantics of the GaussianDistributionType:

Name	Definition
GaussianDistributionType	Describes a continuous multi-dimensional probability density function that takes the form of a Gaussian distribution by specifying the parameters of the distribution. Parameters of the Gaussian distribution such as mean and variance are included via the base ProbabilityDistributionType. The probability density function for the GaussianDistributionType in the case independent x_i is given by: $f_x(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \text{ for } -\infty < x < \infty \text{ and } \sigma^2 > 0.$ Here, μ gives the mean and σ gives the standard deviation. In order to describe the Gaussian distribution, the values of mean and variance shall be declared. GaussianDistributionType extends ContinuousDistributionType.
CovarianceInverse	Describes the covariance parameter that characterizes the Gaussian distribution by describing the inverse covariance matrix Σ^{-1} (optional). If the value of CovarianceInverse is not specified, it is assumed to have a value equal to the identity matrix of dimension = dim, and the above equation holds for the Gaussian distribution. Otherwise, the probability density function for the Gaussian distribution is given as follows: $f_x(x) = \frac{1}{\sqrt{2\pi \det(\Sigma)^{1/2}}} \exp\left[-\frac{(x-\mu)\Sigma^{-1}(x-\mu)^t}{2}\right], \text{ for } -\infty < x < \infty$ and $\sigma^2 > 0$, where Σ gives the covariance matrix.
Determinant	Describes the expression $\det(\Sigma)$ in the equation for the Gaussian distribution (optional).

Semantics of the GeneralizedGaussianDistributionType:

Name	Definition

Name	Definition
GeneralizedGaussianDistributionType	Describes a continuous multi-dimensional probability density function that takes the form of a generalized Gaussian distribution by specifying the shape parameter of the distribution. GeneralizedGaussianDistributionType extends GaussianDistributionType.
Shape	Describes the shape parameter that specifies the form of the generalized Gaussian distribution. The Shape is specified as a vector in which the value of each element of the vector indicates the Shape along one dimension of the generalized Gaussian distribution. The probability density function for the generalized Gaussian distribution is given as follows: $f_x(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{ (x-\mu) ^r}{2\sigma^2}\right], \text{ for } -\infty < x < \infty \text{ and } \sigma^2 > 0,$ where μ corresponds to the mean, σ corresponds to the standard deviation, and r corresponds to the shape.

Semantics of the ExponentialDistributionType:

Name	Definition
ExponentialDistributionType	Describes a continuous multi-dimensional probability density function that takes the form of an exponential distribution by specifying the parameter of the distribution. The probability density function for the exponential distribution is given as follows: $f_x(x) = \Theta e^{-\Theta x}$, for $x > 0$, where $\Theta > 0$ specifies the parameter of the distribution. ExponentialDistributionType extends ContinuousDistributionType.
Theta	Describes the Theta parameter $\Theta > 0$ that specifies the form of the exponential distribution.

Semantics of the GammaDistributionType:

Name	Definition
GammaDistributionType	Describes a continuous multi-dimensional probability density function that takes the form of a gamma distribution by specifying the parameters of the distribution. The probability density function for the gamma distribution is given as follows: $f_x(x) = \frac{\Theta^r x^{r-1} e^{-\Theta x}}{\Gamma(r)}$, for $x \geq 0$, where $\Theta > 0$ and $r > 0$ specify the parameters of the distribution and $\Gamma(r)$ is the gamma function. Except in special cases, the cumulative density function of the gamma distribution with parameters Θ and r need to be determined from standard tables of the cumulative density function of the standard gamma distribution with parameter r . GammaDistributionType extends ContinuousDistributionType.
Theta	Describes the Theta parameter $\Theta > 0$ that specifies the form of the

Name	Definition
	gamma distribution.
Shape	Describes the Shape parameter $r > 0$ that specifies the form of the gamma distribution.

Semantics of the ContinuousUniformDistributionType:

Name	Definition
ContinuousUniformDistributionType	Describes a continuous multi-dimensional probability density function that takes the form of a continuous uniform distribution by specifying the parameters of the distribution. The probability density function for the continuous uniform distribution is given as follows: $f_x(x) = \frac{1}{b-a}$, for $a \leq x \leq b$, where $-\infty < a < b < \infty$ specify the range parameters of the distribution. ContinuousUniformDistributionType extends ContinuousDistributionType.
MaxRange	Describes the MaxRange parameter $b \geq x$ that gives the maximum value of the range of the continuous uniform distribution.
MinRange	Describes the MinRange parameter $a \leq x$ that gives the minimum value of the range of the continuous uniform distribution.

Semantics of the LognormalDistributionType:

Name	Definition
LognormalDistributionType	Describes a continuous multi-dimensional probability density function that takes the form of a lognormal distribution by specifying the parameter of the distribution. The probability density function for the lognormal distribution is given as follows: $f_x(x) = \frac{1}{\sqrt{2\pi x^2 \delta^2}} \exp\left[-\frac{(\log x - \zeta)^2}{2\delta^2}\right]$, for $x > 0$. Here, $-\infty < \zeta < \infty$ and $\delta^2 > 0$ specify the parameters of the distribution. LognormalDistributionType extends ContinuousDistributionType.
NormalMean	Describes the NormalMean parameter $-\infty < \zeta < \infty$ that gives the mean of the normal distribution that produces the lognormal distribution.
NormalVariance	Describes the NormalVariance parameter $\delta^2 > 0$ that gives the variance of the normal distribution that produces the lognormal distribution.

Semantics of the GaussianMixtureModelType:

Name	Definition
GaussianMixtureModelType	Describes a Gaussian mixture model as a special case of a ContinuousDistributionType. The values of the distribution are determined from the weighted sum or mixture of Gaussian distributions that characterizes the probability of observations of the states. GaussianMixtureModelType extends ContinuousDistributionType.
Weight	Describes the weighting to apply to each Gaussian distribution to form the mixture. Weight is specified as a double.
GaussianDistribution	Describes the continuous Gaussian distribution.

14.4.5.4 Continuous distribution description tools examples (informative)

The following example illustrates the use of the GaussianDistribution DS for describing a one-dimensional Gaussian distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="GaussianDistributionType" dim="1">
    <Mean mpeg7:dim="1"> 0.5 </Mean>
    <Variance mpeg7:dim="1"> 0.25 </Variance>
  </DescriptionUnit>
</Mpeg7>
```

The following example illustrates the use of the ExponentialDistribution DS for describing a two-dimensional generalized Gaussian distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="GeneralizedGaussianDistributionType" dim="2">
    <Mean mpeg7:dim="2"> 0.5 0.35 </Mean>
    <Variance mpeg7:dim="2"> 0.25 0.75 </Variance>
    <Shape mpeg7:dim="2"> 2 2 </Shape>
  </DescriptionUnit>
</Mpeg7>
```

The following example illustrates the use of the ExponentialDistribution DS for describing a two-dimensional exponential distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="ExponentialDistributionType" dim="2">
    <Theta mpeg7:dim="2"> 0.5 0.25 </Theta>
  </DescriptionUnit>
</Mpeg7>
```

The following example illustrates the use of the GammaDistribution DS for describing a three-dimensional Gamma distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="GammaDistributionType" dim="3">
    <Theta mpeg7:dim="3"> 0.4 0.3 0.75 </Theta>
    <Shape mpeg7:dim="3"> 8 4 16 </Shape>
  </DescriptionUnit>
```

```
</Mpeg7>
```

The following example illustrates the use of the ContinuousUniformDistribution DS for describing a one-dimensional continuous uniform distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="ContinuousUniformDistributionType" dim="1">
    <MaxRange mpeg7:dim="1"> 1.0 </MaxRange>
    <MinRange mpeg7:dim="1"> 0.0 </MinRange>
  </DescriptionUnit>
</Mpeg7>
```

The following example illustrates the use of the LognormalDistribution DS for describing a one-dimensional lognormal distribution.

```
<Mpeg7>
  <DescriptionUnit xsi:type="LognormalDistributionType" dim="1">
    <NormalMean mpeg7:dim="1"> 0.5 </NormalMean>
    <NormalVariance mpeg7:dim="1"> 0.35 </NormalVariance>
  </DescriptionUnit>
</Mpeg7>
```

14.4.6 Finite state model description tools

14.4.6.1 Introduction

This subclause specifies models of Markov sources or equivalently, probabilistic functions of Markov chains. The underlying assumption of finite state models is that a system may be described at any time as being in one of a finite set of distinct states. For example, the states may correspond to the different program topics of a news programs, i.e., local news, weather, sports, commercials, and so forth. At discrete times, the system undergoes a change of state, which is possibly back to the same state, according to a set of probabilities associated with the state. For example, a probability is associated with the likelihood of the transition from sports to weather in news programs. In the special case of discrete-time, first order Markov chains, the probabilistic dependence between states is truncated such that the probability of the current state depends only on the preceding state. Furthermore, it is assumed that the probabilities are independent of time, which leads to a set of state-transition probabilities.

The tools in this subclause describe finite state models, state transition models and hidden Markov models. The assumption made for State Transition Models is that each state corresponds to a deterministically observable event, that is, the output produced by a source in any given state is not random. Since this model is often too restrictive, the hidden Markov models includes the case in which the observation is a probabilistic function of the state.

FiniteStateModelType is an abstract type that forms the base type of the different specialized types of finite state models such as StateTransitionModelType, DiscreteHiddenMarkovModelType and ContinuousHiddenMarkovModelType. The FiniteStateModelType includes an attribute that indicates the number of states, which applies to StateTransitionModelType, DiscreteHiddenMarkovModelType and ContinuousHiddenMarkovModelType. The StateTransitionModelType describes each of the states and gives both the state-transition probabilities as a matrix and the initial state probabilities as a vector. The DiscreteHiddenMarkovModelType and the ContinuousHiddenMarkovModelType further extend the StateTransitionModelType and provide additional attributes and elements that describe the probabilistic nature of the different observations of each state. The DiscreteHiddenMarkovModelType includes an attribute that indicates the number of observations per state. It also specifies a sequence that defines the observation symbols and the observation symbol probability distribution of each State. The ContinuousHiddenMarkovModelType specifies a sequence of probability distribution that describes the characteristics of observed descriptor values in each state.

14.4.6.2 Finite state model description tools syntax

```

<!-- ##### -->
<!-- Definition of FiniteStateModel DS (14.4.6) -->
<!-- ##### -->

<!-- Definition of FiniteStateModel DS -->
<complexType name="FiniteStateModelType" abstract="true">
  <complexContent>
    <extension base="mpeg7:ProbabilityModelType">
      <attribute name="numOfStates" type="nonNegativeInteger" use="required"/>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of StateTransitionModel DS -->
<complexType name="StateTransitionModelType">
  <complexContent>
    <extension base="mpeg7:FiniteStateModelType">
      <sequence>
        <element name="Initial" type="mpeg7:ProbabilityMatrixType"
          minOccurs="0"/>
        <element name="Transitions" type="mpeg7:ProbabilityMatrixType"/>
        <element name="State" type="mpeg7:ModelStateType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of DiscreteHiddenMarkovModel DS -->
<complexType name="DiscreteHiddenMarkovModelType">
  <complexContent>
    <extension base="mpeg7:StateTransitionModelType">
      <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="NumOfObservationsPerState"
          type="nonNegativeInteger"/>
        <element name="Observation" type="mpeg7:TextualType" minOccurs="1"
          maxOccurs="unbounded"/>
        <element name="ObservationDistribution"
          type="mpeg7:DiscreteDistributionType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of ContinuousHiddenMarkovModel DS -->
<complexType name="ContinuousHiddenMarkovModelType">
  <complexContent>
    <extension base="mpeg7:StateTransitionModelType">
      <sequence>
        <element name="DescriptorModel" type="mpeg7:DescriptorModelType"
          minOccurs="1" maxOccurs="unbounded"/>
        <element name="ObservationDistribution"
          type="mpeg7:ContinuousDistributionType" minOccurs="1"
          maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

14.4.6.3 Finite state model description tools semantics

Semantics of the FiniteStateModelType:

Name	Definition
FiniteStateModelType	Describes a finite state model (abstract). FiniteStateModelType extends ProbabilityModelType.
numOfStates	Indicates the number of states in the finite state model.

Semantics of the StateTransitionModelType:

Name	Definition
StateTransitionModelType	Describes a state transition model by giving the states and the state-transition probabilities and initial state probabilities. The StateTransitionModelType describes states that correspond to deterministically observable events. StateTransitionModelType extends FiniteStateModelType.
Initial	Describes the initial probabilities the states. There is a one-to-one correspondence between the elements of the initial probability vector and the states.
Transitions	Describes the state-transition probabilities among the states. There is a correspondence between the states and the elements along each dimension of the state-transition probability matrix. The meaning of each element in the transition matrix X is as following: $X = [x_{ij}] (i = 1, 2, \dots, N, j = 1, 2, \dots, N)$ where x_{ij} is the transition probability from state i to state j, and N is equal to value of numOfStates. The input states are defined along the rows of the matrix and the output states are defined along the columns of the matrix. For example, the transition probability from state 1 to state 2 is defined by $X[1,2]$.
State	Describes the sequence of states. The State shall be ordered to correspond to the elements of the Initial probabilities vector and state-transition probabilities matrix.

Semantics of the DiscreteHiddenMarkovModelType:

Name	Definition
DiscreteHiddenMarkovModelType	Describes a discrete hidden Markov model as a special case of state transition model in which the states correspond to randomly observable events. The outputs of the states are specified as observations of which each state is assigned an observation distribution that characterizes the probability of each observation of each state. DiscreteHiddenMarkovModelType extends FiniteStateModelType.
NumOfObservationsPerState	Describes the number of distinct observation symbols per state, for example, the size of the discrete alphabet of the state. The observation symbols correspond to the physical output of the system being modeled. Each state does not need to have an equal number of observations.
Observation	Describes a sequence of the symbols that correspond to the distinct observations of the state. There is a correspondence of the symbol elements of the sequence and the elements of the discrete observation distribution.
ObservationDistribution	Describes the discrete distribution of the observation symbols of the state. There is a one-to-one correspondence between the elements of the in observation distribution and the observations of the state, which is given by the order of the elements and observations.

Semantics of the ContinuousHiddenMarkovModelType:

Name	Definition
ContinuousHiddenMarkovModelType	Describes a continuous hidden Markov model as a special case of state transition model in which a continuous probability distribution is associated with each state. ContinuousHiddenMarkovModelType extends FiniteStateModelType.
DescriptorModel	Describes a template of an observed descriptor by selecting elements from the descriptor. The template is formed by describing the mapping of elements of the descriptor to the fields of the observation distribution. This element can appear more than once to accommodate the case when the observation distribution is composed by multiple features. In this case, the selected elements of each DescriptorModel are concatenated by the order in which they appear in the description.
ObservationDistribution	Describes the continuous probability distribution associated with each state. There shall be one observation distribution for each state defined in the ContinuousHiddenMarkovModelType. The mapping of ObservationDistribution to States is given by the order in which they appear in the description. For example, the second instance of an ObservationDistribution maps to the second instance of a States. Each ObservationDistribution describes a continuous probability distribution that models features of the data. The types of modeled features are described by the DescriptorModel.

14.4.6.4 Finite state model description tools examples (informative)

The following example shows the use of the StateTransitionModel DS for describing a state-transition model that has three states. In this example, assume that the observed weather has one of the following states: "precipitation", "cloudy", or "sunny". Furthermore, the state-transition probabilities indicate the probability of moving from one state to another. The initial probabilities specify the initial probability of each state. The state-transition model allows questions to be answered such as, what is the probability that the weather for eight consecutive days is "sun-sun-sun-rain-rain-sun-cloudy-sun", or given that the system is in a known state, i.e., "sunny", what is the expected number of consecutive days that the system remains in that state.

```
<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="StateTransitionModelType" numOfStates="3">
      <Initial mpeg7:dim="3"> 0.5 0.25 0.25 </Initial>
      <Transitions mpeg7:dim="3 3"> 0.4 0.3 0.3 0.2 0.6 0.2 0.1 0.1 0.8
    </Transitions>
    <State>
      <Label>
        <Name>Precipitation </Name>
      </Label>
    </State>
    <State>
      <Label>
        <Name>Cloudy </Name>
      </Label>
    </State>
    <State>
      <Label>
        <Name>Sunny </Name>
      </Label>
    </State>
  </Model>
</Description>
</Mpeg7>
```

The following example shows the use of the StateTransitionModel DS for describing a state-transition model of the events depicted a video of a soccer game. The model describes three states: "Pass", "Shot on goal" and "Goal score". The StateTransitionModel DS describes the states, the initial state probabilities, and the transition probabilities between the states.

```
<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="StateTransitionModelType" numOfStates="3">
      <Initial mpeg7:dim="3"> 0.25 0.5 0.25 </Initial>
      <Transitions mpeg7:dim="3 3"> 0.2 0.2 0.6 0.1 0.8 0.1 0.3 0.3 0.4
    </Transitions>
    <State>
      <Label>
        <Name> Pass </Name>
      </Label>
    </State>
    <State>
      <Label>
        <Name>Shot on goal </Name>
      </Label>
    </State>
    <State>
      <Label>
        <Name>Goal score </Name>
      </Label>
    </State>
  </Model>
</Description>
</Mpeg7>
```

```

        </Label>
    </State>
</Model>
</Description>
</Mpeg7>

```

The following example shows the use of `DiscreteHiddenMarkovModelType` for describing a discrete hidden markov model of events in a soccer game, such as "passes" and "goal scores".

```

<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="DiscreteHiddenMarkovModelType" numOfStates="2">
      <Initial mpeg7:dim="2"> 0.5 0.5 </Initial>
      <Transitions mpeg7:dim="2 2"> 0.4 0.6 0.3 0.7 </Transitions>
      <State>
        <Label>
          <Name> Pass </Name>
        </Label>
      </State>
      <State>
        <Label>
          <Name> Goal score </Name>
        </Label>
      </State>
      <NumOfObservationsPerState>2</NumOfObservationsPerState>
      <Observation> Team A pass </Observation>
      <Observation> Team B pass </Observation>
      <ObservationDistribution xsi:type="HistogramProbabilityType"
        momentNormalized="1" dim="2">
        <Distribution mpeg7:dim="2"> 0.2 0.8 </Distribution>
      </ObservationDistribution>
      <NumOfObservationsPerState>2</NumOfObservationsPerState>
      <Observation> Team A goal </Observation>
      <Observation> Team B goal </Observation>
      <ObservationDistribution xsi:type="HistogramProbabilityType"
        momentNormalized="1" dim="2">
        <Distribution mpeg7:dim="2"> 0.4 0.6 </Distribution>
      </ObservationDistribution>
    </Model>
  </Description>
</Mpeg7>

```

The following example shows the use of the `ContinuousHiddenMarkovModel DS` for describing a continuous hidden Markov model of audio sound effects. Each continuous hidden Markov model has five states and represents a sound effect class. The parameters of the continuous density state model can be estimated via training, for example, using the Baum-Welch algorithm. After training, the continuous HMM model consists of a 5x5 state transition matrix, a 5x1 initial state density matrix, and 5 multi-dimensional Gaussian distributions defined in terms of the mean and variance parameters. Each multi-dimensional Gaussian distribution has six dimensions corresponding to audio features described by the `AudioSpectrumFlatness D` (see ISO/IEC 15938-4).

```

<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="ContinuousHiddenMarkovModelType" numOfStates="5">
      <Initial mpeg7:dim="5"> 0.1 0.2 0.1 0.4 0.2 </Initial>
      <Transitions mpeg7:dim="5 5">
        0.2 0.2 0.6 0.0 0.0
        0.1 0.2 0.1 0.3 0.3
        0.4 0.2 0.1 0.1 0.2

```



```

0.2 0.1 0.4 0.2 0.1
0.0 0.2 0.1 0.3 0.4
</Transitions>
<State>
  <Label>
    <Name> State 1 </Name>
  </Label>
</State>
<State>
  <Label>
    <Name> State 2 </Name>
  </Label>
</State>
<State>
  <Label>
    <Name> State 3 </Name>
  </Label>
</State>
<State>
  <Label>
    <Name> State 4 </Name>
  </Label>
</State>
<State>
  <Label>
    <Name> State 5 </Name>
  </Label>
</State>
<DescriptorModel>
  <Descriptor xsi:type="AudioSpectrumFlatnessType" loEdge="250"
    hiEdge="1600">
    <Vector> 1 2 3 4 5 6 </Vector>
  </Descriptor>
  <Field>Vector</Field>
</DescriptorModel>
<ObservationDistribution xsi:type="GaussianDistributionType" dim="6">
  <Mean mpeg7:dim="6"> 0.5 0.5 0.25 0.3 0.5 0.3 </Mean>
  <Variance mpeg7:dim="6"> 0.25 0.75 0.5 0.45 0.75 0.3</Variance>
</ObservationDistribution>
<ObservationDistribution xsi:type="GaussianDistributionType" dim="6">
  <Mean mpeg7:dim="6"> 0.25 0.4 0.25 0.3 0.2 0.1 </Mean>
  <Variance mpeg7:dim="6"> 0.5 0.25 0.5 0.45 0.5 0.2</Variance>
</ObservationDistribution>
<ObservationDistribution xsi:type="GaussianDistributionType" dim="6">
  <Mean mpeg7:dim="6"> 0.2 0.5 0.35 0.3 0.5 0.5 </Mean>
  <Variance mpeg7:dim="6"> 0.5 0.5 0.5 0.5 0.75 0.5</Variance>
</ObservationDistribution>
<ObservationDistribution xsi:type="GaussianDistributionType" dim="6">
  <Mean mpeg7:dim="6"> 0.5 0.3 0.25 0.2 0.5 0.6 </Mean>
  <Variance mpeg7:dim="6"> 0.5 0.1 0.5 0.25 0.75 0.4</Variance>
</ObservationDistribution>
<ObservationDistribution xsi:type="GaussianDistributionType" dim="6">
  <Mean mpeg7:dim="6"> 0.5 0.15 0.25 0.3 0.5 0.35 </Mean>
  <Variance mpeg7:dim="6"> 0.5 0.75 0.5 0.5 0.75 0.35</Variance>
</ObservationDistribution>
</Model>
</Description>
</Mpeg7>

```

14.5 Analytic models

14.5.1 Introduction

Analytic models describe the association of labels or semantics with collections of multimedia content. The collections may be specified by enumerating the members of the collection, such as by using content collections or descriptor collections, or may be specified using parameterized representations of the collections in the form of probability models. This allows the analytic models to characterize different semantic concepts by models in terms of clusters of multimedia data, example collections of multimedia content descriptions, or probability models.

14.5.2 AnalyticModel DS

14.5.2.1 Introduction

The AnalyticModel DS describes the association of labels or semantics with a collection or probability model, and therefore describes a model of a semantic concept. Multiple semantics may be associated with each AnalyticModel. The AnalyticModel DS specifies the semantics in the form of an unbounded set of labels or semantic descriptions. AnalyticModelType is an abstract type that forms the base type of the different specialized types of analytic models.

14.5.2.2 AnalyticModel DS syntax

```
<!-- ##### -->
<!-- Definition of AnalyticModel DS (14.5.2) -->
<!-- ##### -->

<!-- Definition of AnalyticModel DS -->
<complexType name="AnalyticModelType" abstract="true">
  <complexContent>
    <extension base="mpeg7:ModelType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="Label">
          <complexType>
            <complexContent>
              <extension base="mpeg7:TermUseType">
                <attribute name="relevance" type="mpeg7:zeroToOneType"
                  use="optional"/>
                <attribute name="confidence" type="mpeg7:zeroToOneType"
                  use="optional"/>
                <attribute name="reliability" type="mpeg7:zeroToOneType"
                  use="optional"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="Semantics">
          <complexType>
            <complexContent>
              <extension base="mpeg7:SemanticBaseType">
                <attribute name="relevance" type="mpeg7:zeroToOneType"
                  use="optional"/>
                <attribute name="confidence" type="mpeg7:zeroToOneType"
                  use="optional"/>
                <attribute name="reliability" type="mpeg7:zeroToOneType"
                  use="optional"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

```

        </choice>
        <attribute name="function" use="optional">
            <simpleType>
                <restriction base="string">
                    <enumeration value="described"/>
                    <enumeration value="describing"/>
                </restriction>
            </simpleType>
        </attribute>
    </extension>
</complexContent>
</complexType>

<!-- Definition of ModelState DS -->
<complexType name="ModelStateType">
    <complexContent>
        <extension base="mpeg7:AnalyticModelType"/>
    </complexContent>
</complexType>

```

14.5.2.3 AnalyticModel DS semantics

Semantics of the AnalyticModelType:

Name	Definition
AnalyticModelType	Describes the association of labels or semantics with collections of multimedia content (abstract). AnalyticModelType extends ModelType.
Label	Describes a semantic label for the model given as a text description (optional). Multiple labels may be supplied.
relevance	Identifies the relevance of the semantic label to the analytic model (optional). Relevance is specified on a zero-to-one scale, where zero indicates no relevance and one indicates highest relevance.
confidence	Identifies the confidence of the semantics label (optional). Confidence indicates the degree of confidence of the semantic label considering the likelihood of the label. For example, if a label occurs rarely, it may be assigned a low confidence. Whereas, if a label occurs commonly, it may be assigned a high confidence. A value of 1.0 indicates highest confidence. A value of 0.0 indicates lowest confidence.
reliability	Identifies the reliability of the semantics label (optional). Reliability refers to the accuracy of the values of the semantic label, which may take into account the method of extraction. For example, in some cases the semantic label values that are produced automatically may be assigned lower reliability score than those that are extracted manually. A value of 1.0 indicates highest reliability. A value of 0.0 indicates lowest reliability.
Semantics	Describes a semantics for the model given as a Semantics description (optional). Multiple Semantics descriptions may be supplied.
relevance	Identifies the relevance of the semantic entity to the analytic model (optional). Relevance is specified on a zero-to-one scale, where zero indicates no relevance and one indicates highest relevance.

Name	Definition
confidence	Identifies the confidence of the Semantic entity (optional). Confidence indicates the degree of confidence of the Semantic entity considering the likelihood of the Semantic entity. For example, if a Semantic entity occurs rarely, it may be assigned a low confidence. Whereas, if a Semantic entity occurs commonly, it may be assigned a high confidence. A value of 1.0 indicates highest confidence. A value of 0.0 indicates lowest confidence.
reliability	Identifies the reliability of the Semantic entity (optional). Reliability refers to the accuracy of the values of the Semantic entity, which may take into account the method of extraction. For example, in some cases the Semantic entity values that are produced automatically may be assigned lower reliability score than those that are extracted manually. A value of 1.0 indicates highest reliability. A value of 0.0 indicates lowest reliability.
function	Describes whether the semantics or label is being described by the model, or if the semantics or label is describing the model (optional). For example, in the describing case, the Label="Sunsets" can be associated with a collection of images to describe or annotate the images. For example, in the described case, a collection of images can also be associated with the Label="Sunsets" to indicate that the images describe or annotate the concept represented by the Label. In this case, the images may be thought of as example images of sunsets. If the function is not described then the function of the semantics or label is unknown.

Semantics of the ModelStateType:

Name	Definition
ModelStateType	Describes the state of a source of a finite state model. ModelStateType extends AnalyticModelType.

14.5.2.4 AnalyticModel DS examples (informative)

The following example illustrates the use of the ModelState DS to describe a an analytic model that represents a state in a finite state model. The description gives three labels for the state and gives information about the confidence and reliability of the state analytic model.

```

<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="ModelStateType" confidence="0.9" reliability="0.75">
      <Label relevance="0.5" confidence="0.9" reliability="0.75">
        <Name> Shot on goal </Name>
      </Label>
      <Label relevance="1.0" confidence="0.75" reliability="0.75">
        <Name> Ball block </Name>
      </Label>
      <Label relevance="0.25" confidence="0.5" reliability="0.75">
        <Name> Shot wide </Name>
      </Label>
    </Model>
  </Description>
</Mpeg7>

```

14.5.3 CollectionModel DS

14.5.3.1 Introduction

The CollectionModel DS describes the association of semantics with a collection. The CollectionModel DS allows the association of the same semantics with multiple collections by allowing an unbounded set of collections to be specified. CollectionModelType extends AnalyticModelType.

14.5.3.2 CollectionModel DS syntax

```
<!-- ##### -->
<!-- Definition of CollectionModel DS (14.5.3) -->
<!-- ##### -->

<!-- Definition of CollectionModel DS -->
<complexType name="CollectionModelType">
  <complexContent>
    <extension base="mpeg7:AnalyticModelType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="Collection" type="mpeg7:CollectionType"/>
        <element name="CollectionRef" type="mpeg7:ReferenceType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

14.5.3.3 CollectionModel DS semantics

Semantics of the CollectionModelType:

Name	Definition
CollectionModelType	Describes the association of semantics with a collection or multiple collections. The CollectionModel may contain a mix of different types of collections of multimedia content and concepts within a single description. However, the CollectionModel shall be limited to a single type of descriptor per description. CollectionModelType extends AnalyticModelType.
Collection	Describes the collections that are assigned the labels or semantics of the CollectionModelType.
CollectionRef	Specifies references to the collections that are assigned the semantic labels of the CollectionModelType.

14.5.3.4 CollectionModel DS examples (informative)

The following example illustrates the use of the CollectionModel DS for describing a collection model consisting of a content collection of four images. In this example, the semantic concept is being described by the collection of images. For example, the collection of images describe the concept of "soccer shots on goal". In this example, the description gives a confidence of 0.75 and a reliability of 0.5.

```
<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="CollectionModelType" confidence="0.75" reliability="0.5"
      function="described">
```

```

<Label>
  <Name>Soccer shots on goal</Name>
</Label>
<Collection xsi:type="ContentCollectionType">
  <Content xsi:type="ImageType">
    <Image>
      <MediaLocator xsi:type="ImageLocatorType">
        <MediaUri>soccer1.jpg</MediaUri>
      </MediaLocator>
    </Image>
  </Content>
  <Content xsi:type="ImageType">
    <Image>
      <MediaLocator xsi:type="ImageLocatorType">
        <MediaUri>soccer2.jpg</MediaUri>
      </MediaLocator>
    </Image>
  </Content>
  <Content xsi:type="ImageType">
    <Image>
      <MediaLocator xsi:type="ImageLocatorType">
        <MediaUri>soccer3.jpg</MediaUri>
      </MediaLocator>
    </Image>
  </Content>
  <Content xsi:type="ImageType">
    <Image>
      <MediaLocator xsi:type="ImageLocatorType">
        <MediaUri>soccer4.jpg</MediaUri>
      </MediaLocator>
    </Image>
  </Content>
</Collection>
</Model>
</Description>
</Mpeg7>

```

The following example illustrates the use of the CollectionModel DS for describing a model of a collection of two color descriptions. In this example, the semantic concept "Sunsets" has the function of describing the collection of descriptors.

```

<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="CollectionModelType" confidence="0.75" reliability="0.5"
      function="describing">
      <Label>
        <Name> Sunsets </Name>
      </Label>
      <Collection xsi:type="DescriptorCollectionType">
        <Descriptor 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>
        </Descriptor>
        <Descriptor 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>
        </Descriptor>
      </Collection>
    </Model>
  </Description>

```

```
</Mpeg7>
```

The following example illustrates the use of the `CollectionModel` DS for describing a collection of four concepts. The concept collection has a semantic label: "soccer stadium objects", that is, the semantic concept has the function of describing the collection of concepts. In this example, the confidence is 0.75 and reliability is 0.5.

```
<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="CollectionModelType" confidence="0.75" reliability="0.5"
      function="describing">
      <Label>
        <Name>Soccer stadium objects</Name>
      </Label>
      <Collection xsi:type="ConceptCollectionType">
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Ball </Name>
          </Label>
        </Concept>
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Goal </Name>
          </Label>
        </Concept>
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Seats </Name>
          </Label>
        </Concept>
        <Concept xsi:type="ObjectType">
          <Label>
            <Name> Water </Name>
          </Label>
        </Concept>
      </Collection>
    </Model>
  </Description>
</Mpeg7>
```

14.5.4 DescriptorModel DS

14.5.4.1 Introduction

The `DescriptorModel` DS describes a template or model of a known descriptor by selecting elements from the descriptor. The model is formed by describing the mapping of elements of the descriptor type to the fields of the descriptor model. The `DescriptorModel` DS is used by `ProbabilityModelClass` DS (see 14.5.5) and `ClusterModel` DS (see 14.6) to describe a class or cluster of descriptors using probability models or statistics computed over the aggregation of multiple descriptors.

14.5.4.2 DescriptorModel DS syntax

```
<!-- ##### -->
<!-- Definition of DescriptorModel DS (14.5.4) -->
<!-- ##### -->

<!-- Definition of DescriptorModel DS -->
<complexType name="DescriptorModelType">
```

```

<complexContent>
  <extension base="mpeg7:ModelType">
    <sequence>
      <element name="Descriptor" type="mpeg7:DType"/>
      <element name="Field" type="mpeg7:xPathFieldType"
        minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </extension>
</complexContent>
</complexType>

```

14.5.4.3 DescriptorModel DS semantics

Semantics of the DescriptorModelType:

Name	Definition
DescriptorModelType	Describes a template or model of a known descriptor by selecting elements from the descriptor. The model is formed by describing the mapping of elements of the descriptor to the fields of the descriptor model. DescriptorModelType extends ModelType.
Descriptor	Describes an example instance of the Descriptor in order to allow its elements to be modeled. The Descriptor example shall provide a valid description of the Descriptor including all required attributes and elements. Any number of the elements of the Descriptor can be used to form the descriptor model by specifying the mapping of elements of the descriptor to the fields of the descriptor model. Any elements of the example Descriptor that are not mapped to fields of the descriptor model are assumed to be constant in the descriptor model, taking the values specified in the example Descriptor. All attributes of the example Descriptor are also assumed to be constant in the descriptor model, taking the values specified in the example Descriptor.
Field	Describes the sequence of elements from the example Descriptor that are being modeled. The fields of the model, which correspond to elements of the Descriptor, are described by specifying the XPath expression to the elements of the Descriptor. The XPath locator shall be used only to locate elements in the immediately preceding Descriptor example. Multiple fields can be described, in which case, the elements of the Descriptor are concatenated in the order given by the specification of the fields. The context node of the XPath expression is the node of the preceding Descriptor element.

14.5.4.4 DescriptorModel DS examples (informative)

The following example illustrates the use of the DescriptorModel DS for describing a descriptor model of the ScalableColor descriptor. The DescriptorModel DS specifies that the descriptor model is formed from the Coefficients element of the ScalableColor D (defined in ISO/IEC 15938-3). For example, in the DescriptorModel for ScalableColorType, the value of numOfCoefficients="16" is constant in the descriptor model.

```

<mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="DescriptorModelType">
      <Descriptor 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>
      </Descriptor>
    <Field>Coeff</Field>
  </Model>
</Description>
</Mpeg7>

```

The following example illustrates the use of the DescriptorModel DS for describing a descriptor model of the ContourShape descriptor. The DescriptorModel DS specifies that the descriptor model is formed from the concatenation of six elements of the ContourShape D (defined in ISO/IEC 15938-3). All other elements and attributes not mentioned in the field statements are assumed to be constant in the descriptor models, taking the values specified in the example Descriptors.

```

<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="DescriptorModelType">
      <Descriptor xsi:type="ContourShapeType">
        <GlobalCurvature> 3 5 </GlobalCurvature>
        <PrototypeCurvature> 4 7 </PrototypeCurvature>
        <HighestPeakY>2</HighestPeakY>
        <Peak peakX="4" peakY="2"/>
        <Peak peakX="3" peakY="5"/>
        <Peak peakX="7" peakY="1"/>
      </Descriptor>
      <Field>GlobalCurvature</Field>
      <Field>PrototypeCurvature</Field>
      <Field>HighestPeakY</Field>
      <Field>Peak</Field>
    </Model>
  </Description>
</Mpeg7>

```

14.5.5 ProbabilityModelClass DS

14.5.5.1 Introduction

The ProbabilityModelClass DS describes the association of semantics with classes of descriptions of multimedia content, which are described using probability models. The probability models provide an aggregate statistical representation of the descriptions in a class. The ProbabilityModelClass DS allows the association of the same semantics with multiple probability models by allowing an unbounded set of probability models to be specified. ProbabilityModelClassType extends AnalyticModelType.

14.5.5.2 ProbabilityModelClass DS syntax

```

<!-- ##### -->
<!-- Definition of ProbabilityModelClass DS (14.5.5) -->
<!-- ##### -->

<!-- Definition of ProbabilityModelClass DS -->
<complexType name="ProbabilityModelClassType">
  <complexContent>
    <extension base="mpeg7:AnalyticModelType">
      <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="DescriptorModel" type="mpeg7:DescriptorModelType"/>
        <element name="ProbabilityModel" type="mpeg7:ProbabilityModelType"/>
      </sequence>
    </extension>
  </complexContent>

```

```
</complexType>
```

14.5.5.3 ProbabilityModelClass DS semantics

Semantics of the ProbabilityModelClassType:

Name	Definition
ProbabilityModelClassType	Describes the association of semantics with probability models of multimedia content. ProbabilityModelClassType extends AnalyticModelType.
DescriptorModel	Describes an example instance of the Descriptor in order to allow the elements of the Descriptor to be modeled using a probability model. The DescriptorModel describes the sequence of fields that are concatenated to form a vector. The aggregation of instances of these vectors are then modeled using the probability model.
ProbabilityModel	Describes the probability model of the aggregation of descriptions of type of the DescriptorModel that form the class.

14.5.5.4 ProbabilityModelClass DS examples (informative)

The following example illustrates the use of the ProbabilityModelClass DS for describing a probability model class that characterizes a class of "Nature scene" images by representing the statistics associated with the color features of the images of that class. For example, the following example indicates that the Coefficients element of the ScalableColor D (defined in ISO/IEC 15938-3) for the Nature scene images has a centroid or mean value of (0.5, 0.5, ...) and a variance of (0.25, 0.75, ...).

```
<mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="ProbabilityModelClassType" confidence="0.75"
      reliability="0.5">
      <Label>
        <Name>Nature scenes</Name>
      </Label>
      <DescriptorModel>
        <Descriptor 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>
        </Descriptor>
        <Field>Coeff</Field>
      </DescriptorModel>
      <ProbabilityModel xsi:type="ProbabilityDistributionType" confidence="1.0"
        dim="16">
        <Mean mpeg7:dim="16"> 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 </Mean>
        <Variance mpeg7:dim="16"> 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 5 </Variance>
      </ProbabilityModel>
    </Model>
  </Description>
</mpeg7>
```

The following example illustrates the use of the ProbabilityModelClass DS for describing a probability model that characterizes checker patterns using a probability model of EdgeHistogram D (see ISO/IEC 15938-3).

```
<Mpeg7>
```

```

<Description xsi:type="ModelDescriptionType">
  <Model xsi:type="ProbabilityModelClassType" confidence="0.75"
    reliability="0.5">
    <Label relevance="0.75">
      <Name>Checker patterns</Name>
    </Label>
    <DescriptorModel>
      <Descriptor xsi:type="EdgeHistogramType">
        <BinCounts> 3 5 2 5 . . . 6 </BinCounts>
      </Descriptor>
      <Field>BinCounts</Field>
    </DescriptorModel>
    <ProbabilityModel xsi:type="ProbabilityDistributionType" confidence="0.75"
      dim="80">
      <Mean mpeg7:dim="80"> 5 3 9 4 . . . 5 </Mean>
      <Variance mpeg7:dim="80"> 4.5 5.0 8.0 7.5 . . . 5.2 </Variance>
    </ProbabilityModel>
  </Model>
</Description>
</Mpeg7>

```

The following example illustrates the use of the ProbabilityModelClass DS for describing a probability model that characterizes oval shapes using a probability model of RegionShape D (see ISO/IEC 15938-3).

```

<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="ProbabilityModelClassType" confidence="0.75"
      reliability="0.5">
      <Label relevance="0.75">
        <Name> Ovals </Name>
      </Label>
      <DescriptorModel>
        <Descriptor xsi:type="RegionShapeType">
          <MagnitudeOfART> 3 5 2 5 . . . 6 </MagnitudeOfART>
        </Descriptor>
        <Field>MagnitudeOfART</Field>
      </DescriptorModel>
      <ProbabilityModel xsi:type="ProbabilityDistributionType" confidence="1.0"
        dim="35">
        <Mean mpeg7:dim="35"> 4 8 6 9 . . . 5 </Mean>
        <Variance mpeg7:dim="35"> 1.3 2.5 5.0 4.5 . . . 3.2 </Variance>
      </ProbabilityModel>
    </Model>
  </Description>
</Mpeg7>

```

The following example illustrates the use of the ProbabilityModelClass DS for describing a probability model that characterizes silhouettes using a probability model of ContourShape D (see ISO/IEC 15938-3).

```

<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="ProbabilityModelClassType" confidence="0.75"
      reliability="0.5">
      <Label relevance="0.75">
        <Name> Silhouettes </Name>
      </Label>
      <DescriptorModel>
        <Descriptor xsi:type="ContourShapeType">

```

```

        <GlobalCurvature> 3 5 </GlobalCurvature>
        <PrototypeCurvature> 4 7 </PrototypeCurvature>
        <HighestPeakY>2</HighestPeakY>
        <Peak peakX="4" peakY="2"/>
        <Peak peakX="3" peakY="5"/>
        <Peak peakX="7" peakY="1"/>
    </Descriptor>
    <Field>GlobalCurvature</Field>
    <Field>PrototypeCurvature</Field>
    <Field>HighestPeakY</Field>
    <Field>Peak</Field>
</DescriptorModel>
<ProbabilityModel xsi:type="ProbabilityDistributionType" confidence="1.0"
    dim="11">
    <Mean mpeg7:dim="11"> 3 9 7 4 6 9 1 2 6 8 9</Mean>
    <Variance mpeg7:dim="11"> 3.4 5.4 9.1 1.8 2.3 6.5 7.9 1.2 3.4 3.3 1.0
    </Variance>
</ProbabilityModel>
</Model>
</Description>
</Mpeg7>

```

14.6 Cluster models

14.6.1 Introduction

Cluster Models describe analytic models and their associated probability models. For example, this allows the description of the centroid of a collection of descriptor instances in order to represent a semantic class.

14.6.2 ClusterModel DS

14.6.2.1 Introduction

The ClusterModel DS describes the analytic model and an associated probability model. The ClusterModel DS allows the association of the same semantics with multiple cluster models by allowing an unbounded set of analytic models to be specified. ClusterModelType extends AnalyticModelType.

14.6.2.2 ClusterModel DS syntax

```

<!-- ##### -->
<!-- Definition of ClusterModel DS (14.6.2) -->
<!-- ##### -->

<!-- Definition of ClusterModel DS -->
<complexType name="ClusterModelType">
    <complexContent>
        <extension base="mpeg7:AnalyticModelType">
            <sequence>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="Collection" type="mpeg7:CollectionType"/>
                    <element name="CollectionRef" type="mpeg7:ReferenceType"/>
                </choice>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="ClusterModel" type="mpeg7:ClusterModelType"/>
                    <element name="ClusterModelRef" type="mpeg7:ReferenceType"/>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        <element name="DescriptorModel" type="mpeg7:DescriptorModelType"/>
        <element name="ProbabilityModel" type="mpeg7:ProbabilityModelType"/>
    </sequence>
</sequence>
</extension>
</complexContent>
</complexType>

```

14.6.2.3 ClusterModel DS semantics

Semantics of the ClusterModelType:

Name	Definition
ClusterModelType	Describes the association of labels or semantics, and statistics or probabilities with collections. ClusterModelType extends AnalyticModelType.
Collection	Describes the collection that provides the multimedia content collection, instance or class information.
CollectionRef	Describes the reference to the associated collection that provides the multimedia content collection, instance or class information.
ClusterModel	Describes an unbounded set of child cluster models of the current ClusterModel (optional). This allows nesting of cluster models within cluster models. If ClusterModel is not specified, then the current ClusterModel terminates the cluster model hierarchy.
ClusterModelRef	Describes an unbounded set of references to child cluster models of the current ClusterModel (optional).
DescriptorModel	Describes an example instance of the Descriptor in order to allow the elements of the Descriptor to be modeled using a probability model. The DescriptorModel describes the sequence of fields that are concatenated to form a vector. The aggregation of instances of these vectors are then modeled using the ProbabilityModel.
ProbabilityModel	Describes the associated probability model that provides a parameterized description of the analytic model (optional).

14.6.2.4 ClusterModel DS examples (informative)

The following example illustrates the use of the ClusterModel DS for describing a cluster model consisting of a set of examples of two scalable color descriptions. The cluster models have a semantic label: "Nature scenes". The example describes also the probability model of the ScalableColor D (see ISO/IEC 15938-3).

```

<Mpeg7>
  <Description xsi:type="ModelDescriptionType">
    <Model xsi:type="ClusterModelType" confidence="0.75" reliability="0.5">
      <Label>
        <Name> Nature scenes </Name>
      </Label>
      <Collection xsi:type="DescriptorCollectionType">
        <Descriptor 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>
        </Descriptor>
      </Collection>
    </Model>
  </Description>
</Mpeg7>

```

```

        <Descriptor 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>
        </Descriptor>
    </Collection>
    <DescriptorModel>
        <Descriptor 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>
        </Descriptor>
        <Field>Coeff</Field>
    </DescriptorModel>
    <ProbabilityModel xsi:type="ProbabilityDistributionType" confidence="1.0"
        dim="16">
        <Mean mpeg7:dim="16"> 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 </Mean>
        <Variance mpeg7:dim="16"> 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 5 </Variance>
    </ProbabilityModel>
</Model>
</Description>
</Mpeg7>

```

14.7 Classification models

14.7.1 Introduction

Classification Models describe information about known multimedia content that can be used to classify unknown multimedia content. The output of the classifiers is the association of labels or semantics with the unknown content. The information the classifier uses involves the association of labels or semantics with the known content.

14.7.2 ClassificationModel DS

14.7.2.1 Introduction

The ClassificationModel DS represents a set of analytic models for the purpose of classifying unknown multimedia content. The ClassificationModelType is an abstract type that forms the base type of the different specialized types of classification models. ClassificationModelType provides attributes that describe the completeness and redundancy by which the classification model covers the semantic space, and the number of analytic models that comprise a classification model. ClassificationModelType extends ModelType.

14.7.2.2 ClassificationModel DS syntax

```

<!-- ##### -->
<!-- Definition of ClassificationModel DS (14.7.2) -->
<!-- ##### -->

<!-- Definition of ClassificationModel DS -->
<complexType name="ClassificationModelType" abstract="true">
    <complexContent>
        <extension base="mpeg7:ModelType">
            <attribute name="complete" type="boolean" use="optional" default="true"/>
            <attribute name="redundant" type="boolean" use="optional" default="false"/>
        </extension>
    </complexContent>
</complexType>

```