

Information technology — Multimedia framework (MPEG-21) —

Part 7: Digital Item Adaptation

ICS 35.040

National foreword

This British Standard reproduces verbatim ISO/IEC 21000-7:2004 and implements it as the UK national standard.

The UK participation in its preparation was entrusted to Technical Committee IST/37, Coding of audio, picture, multimedia and hypermedia information, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international publications referred to in this document may be found in the *BSI Catalogue* under the section entitled “International Standards Correspondence Index”, or by using the “Search” facility of the *BSI Electronic Catalogue* or of British Standards Online.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the ISO/IEC title page, pages ii to ix, a blank page, pages 1 to 355 and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 24 November 2004

© BSI 24 November 2004

INTERNATIONAL STANDARD

ISO/IEC 21000-7

First edition
2004-10-15

Information technology — Multimedia framework (MPEG-21) —

Part 7: Digital Item Adaptation

*Technologies de l'information — Cadre multimédia (MPEG-21) —
Partie 7: Adaptation d'article numérique*

Reference number
ISO/IEC 21000-7:2004(E)



AVA Documents
Document Sharing Hub

Contents

Page

Foreword	viii
Introduction	ix
1 Scope	1
1.1 General	1
1.2 Organization of the document	1
1.3 Overview of Digital Item Adaptation	1
1.4 Overview of Digital Item Adaptation tools	2
1.5 Relation between Digital Item Adaptation and other parts of ISO/IEC 21000	3
1.6 Relation between Digital Item Adaptation and ISO/IEC 15938	4
2 Normative references	4
3 Terms, definitions, symbols, and abbreviated terms	5
3.1 Terms and definitions	5
3.1.1 General terms and definitions	5
3.1.2 DIA-specific terms and definition	6
3.2 Symbols and abbreviated terms	7
3.3 Schema documents	8
3.4 Use of prefixes in this specification	8
3.5 URI fragment identifiers for XML documents	9
3.5.1 Syntax of URI fragment identifiers	9
3.5.2 URI fragment identifiers examples	9
4 Schema tools	10
4.1 Introduction	10
4.2 Schema wrapper	10
4.3 Base types	10
4.3.1 Introduction	10
4.3.2 Base types syntax	11
4.3.3 Base types semantics	11
4.4 Description metadata	11
4.4.1 Introduction	11
4.4.2 Description metadata syntax	12
4.4.3 Description metadata semantics	12
4.5 Root elements	12
4.5.1 Introduction	12
4.5.2 Root element syntax	13
4.5.3 Root element semantics	13
4.5.4 Root element examples	14
4.6 Reference	15
4.6.1 Introduction	15
4.6.2 Reference syntax	15
4.6.3 Reference semantics	15
4.6.4 Reference examples	15
5 Low-level datatypes	16
5.1 Introduction	16
5.2 Unsigned integer datatypes	16
5.2.1 Introduction	16
5.2.2 Unsigned integer datatypes syntax	16
5.2.3 Unsigned integer datatypes semantics	18
5.3 Common datatypes	18
5.3.1 Introduction	18

5.3.2	Single value datatypes.....	19
5.3.3	Vector datatypes.....	20
5.3.4	Matrix datatypes	23
5.4	Base stack function.....	25
5.4.1	Introduction.....	25
5.4.2	Base stack function syntax.....	26
5.4.3	Base stack function semantics.....	27
5.5	Argument types and extensions.....	27
5.5.1	Introduction.....	27
5.5.2	Argument types and extensions syntax	28
5.5.3	Argument types and extensions semantics	30
6	Usage environment description tools	33
6.1	Introduction.....	33
6.2	Schema wrapper.....	33
6.3	Usage environment	34
6.3.1	Introduction.....	34
6.3.2	Usage environment syntax.....	34
6.3.3	Usage environment semantics	35
6.4	User characteristics	35
6.4.1	Introduction.....	35
6.4.2	Users.....	35
6.4.3	User.....	36
6.4.4	UserInfo	37
6.4.5	UsagePreferences	38
6.4.6	UsageHistory	39
6.4.7	AudioPresentationPreferences.....	41
6.4.8	DisplayPresentationPreferences	47
6.4.9	ColorPreference.....	48
6.4.10	StereoscopicVideoConversion	51
6.4.11	GraphicsPresentationPreferences	53
6.4.12	ConversionPreference	54
6.4.13	PresentationPriorityPreference	59
6.4.14	FocusOfAttention	63
6.4.15	AuditoryImpairment	68
6.4.16	VisualImpairment.....	70
6.4.17	ColorVisionDeficiency	73
6.4.18	MobilityCharacteristics	76
6.4.19	Destination	80
6.5	Terminal capabilities	81
6.5.1	Introduction.....	81
6.5.2	Terminals.....	82
6.5.3	Terminal.....	82
6.5.4	CodecCapabilities	83
6.5.5	CodecCapability	84
6.5.6	CodecParameter.....	87
6.5.7	Displays.....	89
6.5.8	Display.....	90
6.5.9	DisplayCapabilities.....	91
6.5.10	AudioOutputs.....	94
6.5.11	AudioOutput.....	95
6.5.12	AudioOutputCapabilities	96
6.5.13	UserInteractionInputs	97
6.5.14	UserInteractionInput	98
6.5.15	DeviceClass	102
6.5.16	PowerCharacteristics.....	102
6.5.17	Storages	103
6.5.18	Storage	104
6.5.19	StorageCharacteristics	105
6.5.20	DatalOs	106

6.5.21	DatalO	106
6.5.22	DatalOCharacteristics	107
6.5.23	Benchmarks	108
6.5.24	Benchmark	109
6.5.25	CPUBenchmark	110
6.5.26	ThreeDBenchmark	111
6.5.27	IPMPTools	111
6.5.28	Terminal capabilities examples	113
6.6	Network characteristics	115
6.6.1	Introduction	115
6.6.2	Networks	116
6.6.3	Network	116
6.6.4	NetworkCapability	117
6.6.5	NetworkCondition	118
6.6.6	Network characteristics example	120
6.7	Natural environment characteristics	120
6.7.1	Introduction	120
6.7.2	NaturalEnvironments	121
6.7.3	NaturalEnvironment	121
6.7.4	Location	122
6.7.5	Time	123
6.7.6	AudioEnvironment	124
6.7.7	IlluminationCharacteristics	126
7	Bitstream syntax description link	128
7.1	Introduction	128
7.2	BSDLink syntax	128
7.3	BSDLink semantics	130
7.4	BSDLink example	131
8	Bitstream Syntax Description	131
8.1	Introduction	131
8.1.1	Introduction to BSD-based adaptation	131
8.1.2	Adaptation architecture	132
8.1.3	BSDL and gBS Schema	133
8.1.4	Bitstream Syntax Description transformation	133
8.1.5	Multi-step adaptation	133
8.1.6	Organization of this clause	134
8.2	Bitstream Syntax Description Language	134
8.2.1	Overview	134
8.2.2	Schema hierarchy	136
8.3	Bitstream generation: BSDL-1 and gBS Schema	137
8.3.1	BSDL-1	137
8.3.2	generic Bitstream Syntax Schema	144
8.4	BS Description generation: BSDL-2	155
8.4.1	Introduction	155
8.4.2	Schema for BSDL-2 extensions	157
8.4.3	Constraints on the syntax of BSDL-2 extensions	159
8.4.4	Semantics of BSDL-2 structural extensions	160
8.4.5	Semantics of BSDL-2 datatypes extensions	161
8.4.6	BSDL-2 validity of BS Schemas	161
8.4.7	Run-time constraints during BS Description generation	162
8.4.8	BS Description generation with BintobSD	162
8.4.9	Examples	163
9	Terminal and network quality of service	167
9.1	Introduction	167
9.2	Schema wrapper	167
9.3	AdaptationQoS	168
9.3.1	Introduction	168
9.3.2	AdaptationQoS syntax	169

9.3.3	AdaptationQoS semantics.....	169
9.3.4	AdaptationQoS example.....	169
9.4	AdaptationQoS modules	170
9.4.1	Introduction.....	170
9.4.2	BaseModule syntax.....	170
9.4.3	BaseModule semantics.....	170
9.5	IOPin	170
9.5.1	Introduction.....	170
9.5.2	IOPin syntax.....	170
9.5.3	IOPin semantics.....	171
9.6	UtilityFunction	171
9.6.1	Introduction.....	171
9.6.2	UtilityFunction syntax.....	172
9.6.3	UtilityFunction semantics.....	172
9.6.4	UtilityRank syntax	173
9.6.5	UtilityRank semantics	174
9.6.6	UtilityFunction example	174
9.7	Look-up table	176
9.7.1	Introduction.....	176
9.7.2	LookUpTable syntax	176
9.7.3	LookUpTable semantics	176
9.7.4	Axis syntax.....	177
9.7.5	Axis semantics	177
9.7.6	Content syntax.....	178
9.7.7	Content semantics	178
9.7.8	LookUpTable example	179
9.8	Stack function.....	180
9.8.1	Introduction.....	180
9.8.2	StackFunction syntax	180
9.8.3	StackFunction semantics.....	180
9.8.4	StackFunction example	180
9.9	Switch mechanism	182
9.9.1	Introduction.....	182
9.9.2	BaseSwitchModule syntax	182
9.9.3	BaseSwitchModule semantics	183
9.9.4	UtilityFunctionSwitch syntax	183
9.9.5	UtilityFunctionSwitch semantics	183
9.9.6	LookUpTableSwitch syntax.....	184
9.9.7	LookUpTableSwitch semantics	184
9.9.8	StackFunctionSwitch syntax.....	185
9.9.9	StackFunctionSwitch semantics	185
10	Universal constraints description tools.....	185
10.1	Introduction.....	185
10.2	Universal constraints description syntax.....	187
10.3	Universal constraints description semantics.....	188
10.4	Universal constraints description examples.....	190
11	Metadata adaptability	195
11.1	Introduction.....	195
11.2	MetadataAdaptation syntax.....	195
11.3	MetadataAdaptation semantics	197
11.4	MetadataAdaptation examples.....	199
12	Session mobility tools	205
12.1	Introduction.....	205
12.2	Format of a DI for session mobility	206
12.3	SessionMobility syntax.....	207
12.4	SessionMobility semantics	208
12.5	Reconstructing a session.....	209
12.6	Session transfer example.....	210

13	DIA configuration tools	211
13.1	Introduction	211
13.2	DIAConfiguration syntax	211
13.3	DIAConfiguration semantics	212
Annex A	(normative) Classification Schemes	214
A.1	Introduction	214
A.2	Classification Schemes	214
A.2.1	ThreeDBenchmarkCS	214
A.2.2	AdaptationQoSCS	215
A.2.3	AQoS Rounding MethodCS	248
A.2.4	CPU BenchmarkCS	248
A.2.5	DeviceClassCS	249
A.2.6	GraphicsCodingFormatCS	250
A.2.7	InvariantDimensionCS	251
A.2.8	KeyInputCS	251
A.2.9	LocationTypeCS	251
A.2.10	MediaInformationCS	252
A.2.11	PlaceTypeCS	255
A.2.12	RenderingFormatCS	259
A.2.13	SceneCodingFormatCS	260
A.2.14	SegmentDecompositionInfoCS	262
A.2.15	StackFunctionOperatorCS	264
Annex B	(informative) Usage of the BSDLink tool	270
B.1	Introduction	270
B.2	Overview of the adaptation architecture	270
B.3	Example instances	273
B.3.1	Introduction	273
B.3.2	Usage environment description	274
B.3.3	BSDLink	275
B.3.4	Steering description	276
B.3.5	Further constraining the usage and usage environment of a Digital Item	278
B.3.6	Bitstream Syntax Description	282
B.3.7	BSD transformation	297
B.3.8	Resources	309
Annex C	(informative) Usage of DIA Configuration	310
C.1	Example 1	310
C.2	Example 2	313
C.3	Example 3	314
Annex D	(informative) Address corrections in (g)BS Descriptions for multi-step adaptations	316
Annex E	(informative) UCD-based adaptation decision making engine	317
E.1	Introduction	317
E.2	ADTE architectures	317
E.3	UCD based ADTE processing model	318
Annex F	(informative) Patent Statements	321
Annex G	(informative) Schema	322
G.1	Schema definition for schema tools (clause 4)	322
G.2	Schema definition for usage environment descriptions tools (clause 6)	323
G.3	Schema definition for Terminal and network quality of service tools (clause 8)	349
	Bibliography	355

Foreword

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

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

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

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents.

The ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured the ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights is registered with the ISO and IEC. Information may be obtained from the companies listed in Annex F.

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

ISO/IEC 21000-7 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 21000 consists of the following parts, under the general title *Information technology — Multimedia framework (MPEG-21)*:

- *Part 1: Vision, Technologies and Strategy*
- *Part 2: Digital Item Declaration*
- *Part 3: Digital Item Identification*
- *Part 5: Rights Expression Language*
- *Part 6: Rights Data Dictionary*
- *Part 7: Digital Item Adaptation*

The following parts are under preparation:

- *Part 8: Reference Software*
- *Part 9: File Format*
- *Part 10: Digital Item Processing*
- *Part 11: Evaluation Methods for Persistent Association Technologies*

Introduction

The multimedia industry is increasing at a rapid pace. For this industry, the term content is widely used across different segments and applied in many different ways. For this reason the term is deliberately avoided within the context of ISO/IEC 21000 specifications, where it has been replaced by the defined terms Digital Item, media resource and resource. Of equal importance for the specifications of the multimedia framework is the notion of the User. A User of a system includes all members of the value chain (e.g., creator, rights holders, distributors and consumers of Digital Items).

Access devices, with a large set of differing terminal and network capabilities, are making their way into peoples' lives. Additionally, these access devices are used in different locations and environments: anywhere and at anytime. The Users, however, are currently not given tools to deal efficiently with all the intricacies of this new multimedia usage context.

Solutions with advanced multimedia functionality are becoming increasingly important as individuals are producing more and more digital media, not only for professional use but also for their personal use. All these "resource providers" have many of the same concerns: management, re-purposing based on consumer and device capabilities, protection of rights, protection from unauthorised access/modification, protection of privacy of providers and consumers, etc. For example, it is becoming increasingly difficult to identify and understand the different intellectual property rights that are associated with the elements of multimedia resources. The boundaries between the delivery of audio (music and spoken word), accompanying artwork (graphics), text (lyrics), video (visual) and synthetic spaces will become increasingly blurred. New solutions are required to manage the access and delivery process of these different resource types in an integrated and harmonized way, entirely transparent to the many different Users of multimedia services.

The need of these solutions motivates the initiatives of the ISO/IEC 21000 Multimedia Framework, which aims to enable transparent and augmented use of multimedia resources across a wide range of networks and devices.

This seventh part of ISO/IEC 21000 specifies tools for the adaptation of Digital Items (as specified in ISO/IEC 21000-2).

Information technology — Multimedia framework (MPEG-21) —

Part 7: Digital Item Adaptation

1 Scope

1.1 General

This part of ISO/IEC 21000 specifies the syntax and semantics of tools that may be used to assist the adaptation of Digital Items, i.e., the Digital Item Declaration and resources referenced by the declaration. The tools could be used to satisfy transmission, storage and consumption constraints, as well as Quality of Service management by the various Users. It is important to emphasize that the adaptation engines themselves are non-normative tools of this part of ISO/IEC 21000.

1.2 Organization of the document

This document describes the various Digital Item Adaptation tools specified in part 7 of the ISO/IEC 21000 standard. In the remainder of this part of ISO/IEC 21000, each tool is described by the following subclauses:

- Syntax: Normative specification of the syntax of the tool using XML Schema.
- Semantic: Normative specification of the semantics of the tool and its components.
- Informative examples: Optionally, informative examples illustrating use of the tool.

1.3 Overview of Digital Item Adaptation

The goal of the Terminals and Networks element described in ISO/IEC 21000-1 is to achieve interoperable transparent access to (distributed) advanced multimedia content by shielding Users from network and terminal installation, management and implementation issues. To achieve this goal, the adaptation of Digital Items is required. This concept is illustrated in Figure 1. As shown in this conceptual architecture, a Digital Item is subject to a resource adaptation engine, as well as a description adaptation engine, which together produce the adapted Digital Item.

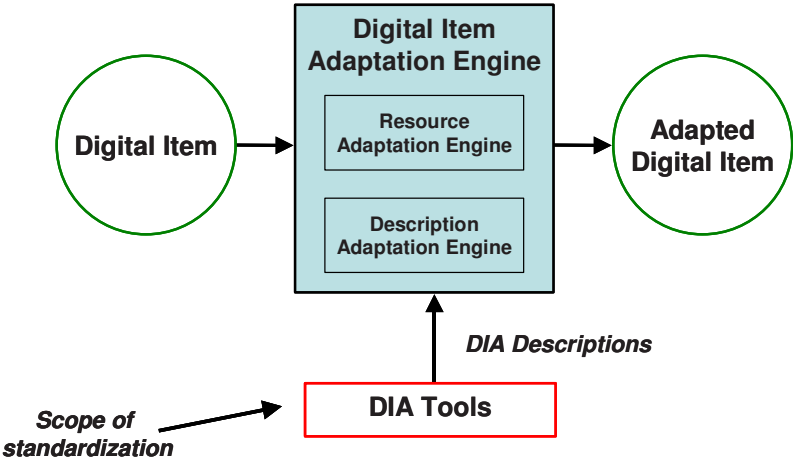


Figure 1 — Illustration of Digital Item Adaptation

It is important to emphasise that the adaptation engines themselves are non-normative tools of Digital Item Adaptation. However, descriptions and format-independent mechanisms that provide support for Digital Item Adaptation in terms of resource adaptation, description adaptation, and/or Quality of Service management are within the scope of the standardization, and are collectively referred to in Figure 1 as DIA Tools.

1.4 Overview of Digital Item Adaptation tools

The Digital Item Adaptation tools in this part of ISO 21000 are clustered into eight major categories as illustrated in Figure 2.

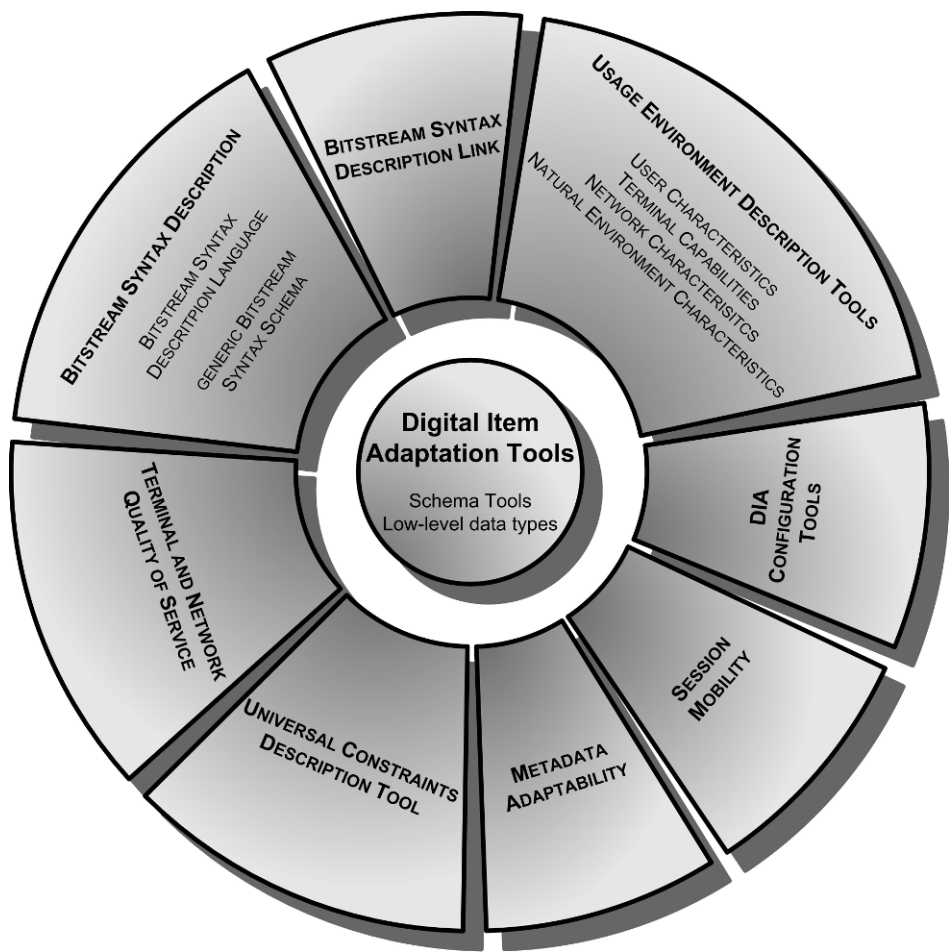


Figure 2 — Overview and organization of Digital Item Adaptation tools

- The categories are clustered according to their functionality and use for Digital Item Adaptation around the *Schema Tools* and *Low-Level Data Types*. The schema tools provide uniform root elements for all DIA descriptions as well as some low-level and basic datatypes which can be used by several DIA tools independently. The syntax and semantics of the schema tools and low-level datatypes are specified in clause 4 and 5, respectively.
- The first major category is the *Usage Environment Description Tools*, which include User characteristics, terminal capabilities, network characteristics and natural environment characteristics. These tools provide descriptive information about the various properties of the usage environment, which originate from Users, to accommodate, for example, the adaptation of Digital Items for transmission, storage and consumption. The syntax and semantics of these tools are specified in clause 6.
- The second category is referred to as *BSDLink* which provides the facilities to create a rich variety of adaptation architectures based on tools specified within this part of ISO/IEC 21000, ISO/IEC 21000-2, and ISO/IEC 15398 among others. The syntax and semantics of this tool is specified in clause 7.
- *Bitstream Syntax Description* tools comprise the third major category of Digital Item Adaptation tools. A BSD describes the syntax – in most cases, the high level structure – of a binary media resource. Using such a description, a Digital Item resource adaptation engine can transform the bitstream and the corresponding description using editing-style operations such as data truncation and simple modifications. These tools are specified in clause 8.
- The fourth category of tools is referred to as *Terminal and Network Quality of Service*. The tools specified in this category describe the relationship between QoS constraints (e.g., on network bandwidth or a terminal's computational capabilities), feasible adaptation operations satisfying these constraints and associated media resource qualities that result from adaptation. This set of tools therefore provides the means to trade-off these parameters with respect to quality so that an adaptation strategy can be formulated and optimal adaptation decisions can be made in constrained environments. The syntax and semantics of these tools are specified in clause 9.
- The *Universal Constraints Description Tools* form the fifth category of tools which enables the possibility to describe limitation and optimisation constraints on adaptations. The syntax and semantics of these tools are specified in clause 10.
- The sixth category is referred to as *Metadata Adaptability*. This tool specifies hint information that can be used to reduce the complexity of adapting the metadata contained in a Digital Item. On the one hand, they are used for filtering and scaling, and on the other hand, for integrating XML instances. The syntax and semantics of this tool are specified in clause 11.
- For *Session Mobility*, the seventh category of tools, the configuration state information that pertains to the consumption of a Digital Item on one device is transferred to a second device. This enables the Digital Item to be consumed on the second device in an adapted way. The syntax and semantics of these tools are specified in clause 12.
- Finally, the eighth category of tools is referred to as *DIA Configuration Tools*, which provides information required for the configuration of a Digital Item Adaptation Engine. The syntax and semantics of these description tools are specified in clause 13.

1.5 Relation between Digital Item Adaptation and other parts of ISO/IEC 21000

The Digital Item is the fundamental unit of distribution and transaction in the Multimedia Framework. While the different parts of ISO/IEC 21000 deal with the components and different aspects of Digital Items, together they form a complete integrated interoperable framework. This subclause describes the relationship of this part of ISO/IEC 21000 with the other parts of ISO/IEC 21000 in addressing the specific function of adapting Digital Items.

ISO/IEC 21000-2 enables the declaration of Digital Items. A Digital Item is a packaging of resources, descriptions and rights expression. A Digital Item may contain elements that conform to ISO/IEC 21000-3,

ISO/IEC 21000-4, ISO/IEC 21000-5, ISO/IEC 21000-6, and tools that are defined in this part of ISO/IEC 21000.

A Digital Item may be input to a Digital Item Adaptation Engine. The Adaptation Engine can modify the input Digital Item by adapting the resources or metadata within the Digital Item or the declaration of the Digital Item to the usage environment. Additionally, the identifiers and rights expressions pertaining to the adapted Digital Item need not be the same as those pertaining to the input Digital Item. This specification deals with adaptation but specifically does not address the relationship of rights and permissions to adaptations. The relationship of rights and permissions is to be addressed in an amendment to this part of ISO/IEC 21000. It is expected that users of this part of ISO/IEC 21000 will register terms describing their specific adaptations with the Registration Authority described in ISO/IEC 21000-6 in order to provide interoperability.

1.6 Relation between Digital Item Adaptation and ISO/IEC 15938

ISO/IEC 15938 is a standard for multimedia content description. For the most part, the description of multimedia content is used to satisfy a User's request for the resources contained in a particular Digital Item. This search would be carried out by a search engine. However, given the Digital Item of interest, ISO/IEC 15938 descriptions could also be used in the adaptation process. For example, ISO/IEC 15938 offers tools for the summarization of media resources, tools that provide transcoding hints about the media resources, and tools that indicate the available variations of a given media resource. See ISO/IEC 15938 for further information on the syntax and semantics of these tools.

Besides serving as an input to the Digital Item Adaptation Engine, several ISO/IEC 15938 tools are also referenced by the DIA specification. For example, tools that indicate a User's preference have been adopted as part of the DIA usage environment description tools, along with tools that indicate location and time associated with a User. The multimedia description schemes specified by ISO/IEC 15938 can also be used to specify the decoding and encoding formats as part of the terminal capabilities description tool. In this particular case, there exists symmetry between tools that are used to describe media resources and tools that are used to describe the capabilities of a terminal. In this way, the media resources can easily be matched or adapted to satisfy the terminal capabilities.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14496-1, *Information technology – Coding of audio-visual objects – Part 1: Systems*

ISO/IEC 14496-2, *Information technology – Coding of audio-visual objects – Part 2: Visual*

ISO/IEC 15938-3, *Information technology – Multimedia content description interface – Part 3: Visual*

ISO/IEC 15938-5, *Information technology – Multimedia Multimedia content description interface – Part 5: Multimedia description schemes*

ISO/IEC 21000 (all parts), *Information technology – Multimedia framework (MPEG-21)*

IEEE 754-1985, IEEE Standard for Binary Floating-Point Arithmetic

Request for Comments (RFC) 1034, *Domain Names – Concepts and Facilities*, The Internet Engineering Task Force (IETF), November 1987

RFC 1738, *Uniform Resource Locators (URL)*, IETF, December 1994

RFC 2141, *URN Syntax*, IETF, May 1997

RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*, IETF, August 1998

Namespaces in XML, W3C Recommendation 14 January 1999

Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation, October 6th, 2000

XSL Transformations (XSLT), Version 1.0 W3C Recommendation, November 16th, 1999

XML Schema Part 1: Structures W3C Recommendation, May 2, 2001

XML Schema Part 2: Datatypes W3C Recommendation, May 2, 2001

XML Information Set, W3C Recommendation 24 October 2001 XML Base, W3C Recommendation 27 June 2001

XML Path Language (XPath), Version 1.0 W3C Recommendation 16 November 1999

XPointer Framework, W3C Recommendation 25 March 2003

XPointer xmlns() Scheme, W3C Recommendation 25 March 2003

XPointer element() Scheme, W3C Recommendation 25 March 2003

3 Terms, definitions, symbols, and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in the following apply:

It should be noted that the term content is widely used across different industries and applied in many different ways. For this reason the term is deliberately avoided within the context of ISO/IEC 21000 specifications, where it has been replaced by the terms Digital Item, media resource and resource, as defined below.

3.1.1 General terms and definitions

3.1.1.1

Description

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

3.1.1.2

Digital Item

A structured digital object conforming to ISO/IEC 21000-2 that is the fundamental unit of transaction and distribution in the multimedia framework.

3.1.1.3

Media resource

A **resource** corresponding to audio-visual or multimedia data.

3.1.1.4

Receiver

The side of a multimedia transaction that receives a **resource**.

3.1.1.5

Resource

A component of a **Digital Item** corresponding to a digital asset or other form of intellectual content.

3.1.1.6

Sender

The side of a multimedia transaction that sends a **resource**.

3.1.1.7

Tool

Definition of syntax and corresponding semantics.

3.1.1.8

User

A user of a multimedia system, which includes all participants of a value network, e.g., creator, rights holders, distributors and consumers of **Digital Items**. A **User** may correspond to a single physical person, as well as a group of people, or an organization.

3.1.2 DIA-specific terms and definition

3.1.2.1

Adaptation unit

Refers to a unit of a logical decomposition of a **bitstream** into segments.

3.1.2.2

Bitstream

A coded **resource** consisting of a structured sequence of binary symbols.

3.1.2.3

Bitstream Syntax

The encoding or format of a **bitstream**.

3.1.2.4

Bitstream Syntax Description

Specification of the high-level structure of a **bitstream** using **Bitstream Syntax Description Language**.

3.1.2.5

Bitstream Syntax Description Language

Language specified in this part of ISO/IEC 21000 for defining a **Bitstream Syntax**.

3.1.2.6

Bitstream Syntax Schema

XML Schema written in the **Bitstream Syntax Description Language** describing the syntax of a given coding representation format.

3.1.2.7

Bitstream Syntax Description Transformation

A process where a **Bitstream Syntax Description** is modified, producing a new description from which an adapted **bitstream** can be generated. This process may be specified for instance by an XSLT style sheet.

3.1.2.8

Digital Item Adaptation

The process of manipulation of a **Digital Item** to produce a modified **Digital Item** in order to adapt it to the **usage environment**.

3.1.2.9

Generic Bitstream Syntax Schema

A generic, coding format independent XML Schema for the description of any binary **media resource**.

3.1.2.10

Resource adaptation

The process of manipulation of a **resource** within a **Digital Item** in order to adapt it to the **usage environment**.

3.1.2.11**Scalable bitstream**

A **bitstream** in which data is organized in such a way that, by retrieving the **bitstream**, it is possible to first render a degraded version of the **resource**, and then progressively improve it by loading additional data.

3.1.2.12**Usage environment description**

Refers to metadata that specifies **User** characteristics, terminal capabilities, network characteristics and natural environment characteristics according to ISO/IEC 21000.

3.2 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviated terms given in the following apply:

BSDL:	Bitstream Syntax Description Language
BSD:	Bitstream Syntax Description
BS Schema:	Bitstream Syntax Schema
CS:	Classification Scheme
DCT:	Discrete Cosine Transform
DI:	Digital Item
DIA:	Digital Item Adaptation
DID:	Digital Item Declaration
DII:	Digital Item Identification
FGS:	Fine Granularity Scalability
IPMP:	Intellectual Property Management & Protection
gBSD:	generic Bitstream Syntax Description
gBS Schema:	generic Bitstream Syntax Schema
RDD:	Rights Data Dictionary
REL:	Rights Expression Language
STX:	Streaming Transformations for XML
UCD:	Universal Constraint Description
UED:	Usage Environment Description
URI:	Uniform Resource Identifier
XML:	Extensible Markup Language 1.0
XSL:	Extensible Stylesheet Language
XSLT:	Extensible Stylesheet Language Transformations

3.3 Schema documents

In the main text of this part of ISO 21000, the syntax of description tools is provided whenever possible as a single schema document.

In some cases though, and in particular for the clauses 4, 6 and 8, the syntax of description tools is provided as a collection of schema snippets imbricated with other text. In order to form a valid schema document, these schema components should be gathered in a same document with the schema wrapper provided at the head of the clause. For better readability, the relevant schema documents are provided in the Annex E, but as non-normative information.

In all cases, each schema document has a `version` attribute, the value of which is "ISO/IEC 21000-7". Furthermore, an informative identifier is given as the value of the `id` attribute of the `schema` component. This identifier is non-normative and used as a convention in this specification to reference another schema document. In particular, it is used for the `schemaLocation` attribute of the `include` and `import` schema components.

Several schema documents defined in this part of ISO/IEC 21000 reference description tools defined in ISO/IEC 15938-5. Therefore, it is necessary to import a corresponding schema document that defines the syntax of these tools. In this part of ISO/IEC 21000, the schema document containing referenced ISO/IEC 15938-5 description tools is identified with the `id` `mpeg7-udp-2003.xsd`.

3.4 Use of prefixes in this specification

For clarity, throughout this part of ISO/IEC 21000, consistent namespace prefixes are used.

"`xsd:`" prefix is not normative. It is a naming convention in this document to refer to an element of the XML Schema namespace (<http://www.w3.org/2001/XMLSchema>). Similarly, "`xsi:`" refers to the <http://www.w3.org/2001/XMLSchema-instance> namespace.

"`xml:`" and "`xmlns:`" are normative prefixes defined in [13]. The prefix "`xml:`" is by definition bound to "<http://www.w3.org/XML/1998/namespace>". The prefix "`xmlns:`" is used only for namespace bindings and is not itself bound to any namespace name.

All other prefixes used in either the text or examples of this specification are not normative, e.g., "`dia:`", "`bs1:`", "`bs2:`", "`sm:`", "`mpeg7:`".

In particular, most of the informative examples in this specification are provided as XML fragments without the normally required XML document declaration and thus miss a correct namespace binding context declaration. In these descriptions fragments, the default namespace is defined as `urn:mpeg:mpeg21:2003:01-DIA-NS` and the different prefixes are bound to the namespaces as given in the following table.

Table 1 — Mapping of prefixes to namespaces in examples and text

Prefix	Corresponding namespace
None (default namespace)	urn:mpeg:mpeg21:2003:01-DIA-NS
dia	urn:mpeg:mpeg21:2003:01-DIA-NS
mpeg7	urn:mpeg:mpeg7:schema:2001
xsi	http://www.w3.org/2001/XMLSchema-instance
xsd	http://www.w3.org/2001/XMLSchema
bs1	urn:mpeg:mpeg21:2003:01-DIA-BSDL1-NS
bs2	urn:mpeg:mpeg21:2003:01-DIA-BSDL2-NS
did	urn:mpeg:mpeg21:2002:01-DIDL-NS
gbsd	urn:mpeg:mpeg21:2003:01-DIA-gBSD-NS
diac	urn:mpeg:mpeg21:2003:01-DIA-DIAC-NS
sm	urn:mpeg:mpeg21:2003:01-DIA-SM-NS

Unlike the informative descriptions examples, the normative specification of the syntax of tools in XML Schema follows the namespace binding context defined in the relevant schema declaration such as the one defined in subclause 4.2. The list of non-normative prefixes provided by Table 1 is not relevant to the schema declaration snippets.

This part of ISO/IEC 21000 normatively references description tools in ISO/IEC 15938 using the `mpeg7` prefix as given Table 1 with the corresponding namespace as defined in ISO/IEC 15938-5:2003. As amendments of ISO/IEC 15938 become available, users of the standard should update the corresponding `mpeg7` namespace. The same remark applies to the namespace defined in ISO/IEC 21000-2 and bound to the `did` prefix given in Table 1.

3.5 URI fragment identifiers for XML documents

3.5.1 Syntax of URI fragment identifiers

Uniform Resource Identifiers (URIs) are used throughout this specification to identify MPEG-21 descriptions or resources. In some cases, URIs are used to identify a fragment of an XML document. For example, it may be necessary to identify a single element, an attribute or a full XML fragment corresponding to the required description information. A specific syntax is then required for such URI fragment identifier.

For this, the XPointer framework [14] is the normative format used in this specification for URI fragment identifiers when the URI is used to locate a fragment of an XML document.

When the URI fragment identifier locates a complex content element, i.e., an XML element containing child sub-elements, then the context of use determines whether it points to the single element or the document fragment below this element.

3.5.2 URI fragment identifiers examples

EXAMPLE 1 The following example shows the declaration of an AdaptationQoS description in a document available at the URI `http://example.com/ex1.xml`. The description itself may be identified by the id of its root element.

```
<DIA xmlns="urn:mpeg:mpeg21:2003:01-DIA-NS">
  <Description xsi:type="AdaptationQoSType" id="myDescrID">
    <Module xsi:type="LookUpTableType">
      <!-- and so on... -->
    </Module>
    <!-- and so on... -->
  </Description>
</DIA>
```

EXAMPLE 2 The following example shows a BSDLink description pointing to the AdaptationQoS description above.

```
<DIA xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS">
  <Description xsi:type="BSDLinkType">
    <SteeringDescriptionRef uri="http://example.com/ex1.xml#myDescrID"/>
    <!-- and so on... -->
  </Description>
</DIA>
```

If the BSDLink and AdaptationQoS descriptions are in the same document, then the URI may be written "#myDescrID".

4 Schema tools

4.1 Introduction

This clause specifies the schema tools that facilitate the creation of descriptions. The following description tools are specified: (1) the base type hierarchy of the description tools defined in this part of ISO/IEC 21000, and (2) the root elements. Base types form the type hierarchy for description tools, while root elements describe the initial wrappers of descriptions. In this clause, we also refer to top-level types, which are types that may follow the root element in a description. The top-level types are defined in the respective clauses in which the related tools to a particular top-level type are defined.

4.2 Schema wrapper

The syntax of description tools specified in this clause is provided as a collection of schema components, consisting notably in type definitions and element declarations. In order to form a valid schema document, these schema components should be gathered in a same document with the following declaration defining in particular the target namespace and the namespaces prefixes.

```
<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for basic schema tools -->
<schema
  version="ISO/IEC 21000-7"
  id="DIA.xsd"
  targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
```

Additionally, the following line should be appended to the resulting schema document in order to obtain a well-formed XML document.

```
</schema>
```

4.3 Base types

4.3.1 Introduction

The following base types are defined in this subclause:

- **DIABaseType**: provides the base abstract type of the type hierarchy. The majority of the tools in this specification are directly defined as an extension of this base type.

- **DIADescriptionType**: **extends** **DIABaseType** and provides the base abstract type for a subset of types defined in this specification that may follow the **DIA** root element as specified in subclause 4.4. Tools that are defined as an extension of **DIADescriptionType** are referred to as top-level types.

4.3.2 Base types syntax

```

<!-- ##### -->
<!-- Definition of DIA Base Types -->
<!-- ##### -->

<complexType name="DIABaseType" abstract="true">
  <complexContent>
    <restriction base="anyType">
      <attribute name="id" type="ID" use="optional"/>
    </restriction>
  </complexContent>
</complexType>

<complexType name="DIADescriptionType" abstract="true">
  <complexContent>
    <extension base="dia:DIABaseType"/>
  </complexContent>
</complexType>

```

4.3.3 Base types semantics

Semantics of the **DIABaseType**:

Name	Definition
DIABaseType	Provides the topmost type of the base type hierarchy. The DIADescriptionType extends DIABaseType .
id	Identifies the id of the DIABaseType . When used as an id for a DIADescriptionType , it shall be used to identify the target referenced by the ReferenceType tool.

Semantics of the **DIADescriptionType**:

Name	Definition
DIADescriptionType	DIADescriptionType extends DIABaseType and provides a base abstract type for a subset of types defined in this specification that may follow the DIA root element as specified in subclause 4.4.

4.4 Description metadata

4.4.1 Introduction

The description metadata provides information such as an alias for URIs identifying a classification scheme.

4.4.2 Description metadata syntax

```
<!-- ##### -->
<!-- Definition of Description Metadata Type -->
<!-- ##### -->

<complexType name="DescriptionMetadataType">
  <complexContent>
    <extension base="dia:DIABaseType">
      <sequence>
        <element name="ClassificationSchemeAlias" maxOccurs="unbounded">
          <complexType>
            <complexContent>
              <extension base="dia:DIABaseType">
                <attribute name="alias" type="NMTOKEN"
                  use="required"/>
                <attribute name="href" type="anyURI" use="required"/>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

4.4.3 Description metadata semantics

Semantics of DescriptionMetadataType:

Name	Definition
DescriptionMetadataType	Tool for describing metadata concerning a DIA description.
ClassificationSchemeAlias	Describes an alias for a classification scheme referenced by a URI.
alias	Describes the alias assigned to the ClassificationScheme. 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 part of ISO/IEC 21000, whether normative or informative, shall be referenced using the identifying value defined by the uri attribute of the ClassificationScheme for that classification scheme.

4.5 Root elements

4.5.1 Introduction

The functionality of a root element is to describe the initial wrapper or root element of descriptions. This subclause specifies two root elements: a DIA root element to be used for complete descriptions, and a DIADescriptionUnit root element to be used for description units. These two forms of descriptions are defined as follows:

- **Complete Description:** describes the target of the Digital Item Adaptation tools using top-level types, which are a subset of types defined in the specification that follow the root element in descriptions. For example, a self-contained description of a usage environment that includes a description of terminal capabilities and a description of User characteristics is a complete description.
- **Description Unit:** describes an instance of a description. A description unit can be used to represent partial information from a complete description. For example, the description of only the display capabilities.

Either the `DIA` or `DIADescriptionUnit` root element shall be used as the topmost element in a description. The use of these root elements must be consistent with the semantics defined below.

4.5.2 Root element syntax

```
<!-- ##### -->
<!-- Definition of DIA Root Element -->
<!-- ##### -->

<element name="DIA">
  <complexType>
    <sequence>
      <element name="DescriptionMetadata" type="dia:DescriptionMetadataType"
        minOccurs="0"/>
      <choice maxOccurs="unbounded">
        <element name="Description" type="dia:DIADescriptionType"/>
        <element name="Reference" type="dia:ReferenceType"/>
      </choice>
    </sequence>
  </complexType>
</element>

<element name="DIADescriptionUnit" type="dia:DIABaseType"/>
```

4.5.3 Root element semantics

Semantics of the `DIA` root element:

Name	Definition
<code>DIA</code>	Serves as the root element for a complete description. <code>DIA</code> shall be used as the topmost element in a complete description.
<code>DescriptionMetadata</code>	Describes the metadata for the descriptions contained within <code>DIA</code> description. The description metadata applies to the elements contained within this description.
<code>Description</code>	Description of the target top-level type according to the specified elements. Only types that are defined as an extension of the <code>DIADescriptionType</code> may follow the <code>DIA</code> root element.
<code>Reference</code>	References an external description of the target top-level type, which is a type that is defined as an extension of the <code>DIADescriptionType</code> .

Semantics of the `DIADescriptionUnit` root element:

Name	Definition
<code>DIADescriptionUnit</code>	Serves as the root element for a description unit. <code>DIADescriptionUnit</code> shall be used as the topmost element in a description unit.

4.5.4 Root element examples

EXAMPLE 1 The following example shows the use of the DIA root element.

```
<DIA>
  <DescriptionMetadata>
    <ClassificationSchemeAlias alias=":ACF:"
      href="urn:mpeg:mpeg7:cs:AudioCodingFormatCS:2001"/>
    <ClassificationSchemeAlias alias=":VCF:"
      href="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001"/>
  </DescriptionMetadata>
  <Description xsi:type="UsageEnvironmentType">
    <UsageEnvironmentProperty xsi:type="TerminalsType">
      <Terminal>
        <TerminalCapability xsi:type="DisplaysType">
          <Display id="d0">
            <DisplayCapability xsi:type="DisplayCapabilityType">
              <Mode>
                <Resolution horizontal="176" vertical="144"/>
              </Mode>
            </DisplayCapability>
          </Display>
        </TerminalCapability>
        <TerminalCapability xsi:type="CodecCapabilitiesType">
          <Decoding xsi:type="AudioCapabilitiesType">
            <Format href=":ACF:4.4">
              <mpeg7:Name xml:lang="en">MP3</mpeg7:Name>
            </Format>
            <Format href=":ACF:6">
              <mpeg7:Name xml:lang="en">AMR</mpeg7:Name>
            </Format>
          </Decoding>
          <Decoding xsi:type="VideoCapabilitiesType">
            <Format href=":VCF:2001:3.1.2">
              <mpeg7:Name xml:lang="en">
                MPEG-4 Visual Simple Profile @ Level 1
              </mpeg7:Name>
            </Format>
          </Decoding>
        </TerminalCapability>
      </Terminal>
    </UsageEnvironmentProperty>
    <UsageEnvironmentProperty xsi:type="NetworksType">
      <Network id="n0">
        <NetworkCharacteristic xsi:type="NetworkCapabilityType"
          maxCapacity="64000"/>
      </Network>
    </UsageEnvironmentProperty>
  </Description>
</DIA>
```

EXAMPLE 2 The following example shows the use of the DIADescriptionUnit root element.

```
<DIADescriptionUnit xsi:type="DisplayCapabilityType">
  <Mode>
    <Resolution horizontal="176" vertical="144"/>
  </Mode>
</DIADescriptionUnit>
```

4.6 Reference

4.6.1 Introduction

This subclause specifies a tool for referencing.

4.6.2 Reference syntax

```
<!-- ##### -->
<!-- Definition of Reference -->
<!-- ##### -->

<complexType name="ReferenceType">
  <attribute name="uri" type="anyURI" use="required"/>
</complexType>
```

4.6.3 Reference semantics

Semantics of ReferenceType:

Name	Definition
ReferenceType	Tool for providing a means to reference elements whose type is derived from DIADescriptionType.
uri	Describes a URI.

4.6.4 Reference examples

EXAMPLE 1 The following example shows the use of `Reference` as an external reference, where the descriptions for a `Terminal` capabilities and `Network` characteristics are stored in another location. In this particular example, only the `Network` element is being referenced through its respective `id` attribute.

```
--Example1.xml--
<DIA>
  <Reference uri="Terminal_and_Network1.xml#n0"/>
</DIA>

--Terminal_and_Network1.xml--
<DIA>
  <Description xsi:type="UsageEnvironmentType">
    <UsageEnvironment xsi:type="TerminalsType">
      <Terminal>
        <TerminalCapability xsi:type="DisplaysType">
          <Display id="d0">
            <DisplayCapability xsi:type="DisplayCapabilityType">
              <Mode>
                <Resolution horizontal="176" vertical="144"/>
              </Mode>
            </DisplayCapability>
          </Display>
        </TerminalCapability>
      </Terminal>
    </UsageEnvironmentProperty>
    <UsageEnvironmentProperty xsi:type="NetworksType">
      <Network id="n0">
        <NetworkCharacteristic xsi:type="NetworkCapabilityType"
          maxCapacity="64000"/>
      </Network>
    </UsageEnvironmentProperty>
  </Description>
</DIA>
```

EXAMPLE 2 When used in the context of a DID, as specified in ISO/IEC 21000-2, it is preferable to use a DID Reference element. The following example shows the use of DID Reference to achieve similar result as that in the previous example.

```
--Example2.xml--
<DIDL>
  <Item>
    <Descriptor>
      <Reference target="Terminal_and_Network2.xml#n0"/>
    </Descriptor>
  </Item>
</DIDL>

--Terminal_and_Network2.xml--
<DIDL>
  <Item>
    <Descriptor id="n0">
      <Statement>
        <DIADescriptionUnit xsi:type=" NetworkCapabilityType"
          maxCapacity="64000"/>
      </Statement>
    </Descriptor>
  </Item>
</DIDL>
```

5 Low-level datatypes

5.1 Introduction

This clause specifies a list of low-level datatypes that may be used throughout this part of ISO/IEC 21000.

5.2 Unsigned integer datatypes

5.2.1 Introduction

This subclause specifies a list of unsigned integer datatypes with no target namespace. This schema inherits the namespace of the including schema, and hence the potential specific semantics associated to the namespace of the including schema.

5.2.2 Unsigned integer datatypes syntax

```
<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Unsigned Integer Datatypes Schema -->
<xsd:schema
  version="ISO/IEC 21000-7"
  id="UnsignedIntegers.xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:simpleType name="b16">
    <xsd:restriction base="xsd:unsignedShort"/>
  </xsd:simpleType>
  <xsd:simpleType name="b15">
    <xsd:restriction base="xsd:unsignedShort">
      <xsd:maxExclusive value="32768"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="b14">
    <xsd:restriction base="xsd:unsignedShort">
      <xsd:maxExclusive value="16834"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="b13">
    <xsd:restriction base="xsd:unsignedShort">
```

```

        <xsd:maxExclusive value="8192"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b12">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="4096"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b11">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="2048"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b10">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="1024"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b9">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="512"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b8">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="256"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b7">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="128"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b6">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="64"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b5">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="32"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b4">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="16"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b3">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="8"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b2">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="4"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="b1">
      <xsd:restriction base="xsd:unsignedShort">
        <xsd:maxExclusive value="2"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:schema>

```

5.2.3 Unsigned integer datatypes semantics

Semantics of the unsigned integer datatypes:

<i>Name</i>	<i>Definition</i>
b1	A 1-bit unsigned integer value.
b2	A 2-bit unsigned integer value.
b3	A 3-bit unsigned integer value.
b4	A 4-bit unsigned integer value.
b5	A 5-bit unsigned integer value.
b6	A 6-bit unsigned integer value.
b7	A 7-bit unsigned integer value.
b8	A 8-bit unsigned integer value.
b9	A 9-bit unsigned integer value.
b10	A 10-bit unsigned integer value.
b11	A 11-bit unsigned integer value.
b12	A 12-bit unsigned integer value.
b13	A 13-bit unsigned integer value.
b14	A 14-bit unsigned integer value.
b15	A 15-bit unsigned integer value.
b16	A 16-bit unsigned integer value.

5.3 Common datatypes**5.3.1 Introduction**

This subclause specifies the syntax and semantics of the various datatypes. These datatypes have been designed in such a way that additional datatypes can easily be added in future versions. In this version, four datatypes are included, namely integers, floats, tokens, and booleans. These datatypes can be used as a single value, a vector or a matrix of values.

5.3.2 Single value datatypes

5.3.2.1 Introduction

This subclause specifies syntax and semantics of single value datatypes.

5.3.2.2 Single value datatypes syntax

```
<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Single Value Data Types -->
<schema
  version="ISO/IEC 21000-7"
  id="SingleValueDatatypes.xsd"
  targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <include schemaLocation="DIA.xsd"/>

  <!-- ##### -->
  <!-- Definition of Single Value Data Types -->
  <!-- ##### -->

  <complexType name="ValueDataType" abstract="true">
    <complexContent>
      <extension base="dia:DIABaseType"/>
    </complexContent>
  </complexType>

  <complexType name="IntegerType">
    <complexContent>
      <extension base="dia:ValueDataType">
        <sequence>
          <element name="Value" type="integer"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="FloatType">
    <complexContent>
      <extension base="dia:ValueDataType">
        <sequence>
          <element name="Value" type="float"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="TokenType">
    <complexContent>
      <extension base="dia:ValueDataType">
        <sequence>
          <element name="Value">
            <simpleType>
              <restriction base="token">
                <pattern value="\c+"/>
              </restriction>
            </simpleType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

```

        </simpleType>
      </element>
    </sequence>
  </extension>
</complexContent>
</complexType>

<complexType name="BooleanType">
  <complexContent>
    <extension base="dia:ValueDataType">
      <sequence>
        <element name="Value" type="boolean"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

</schema>
```

5.3.2.3 Single value datatypes semantics

Semantics of the ValueDataType, IntegerType, FloatType, TokenType, and BooleanType:

Name	Definition
ValueDataType	ValueDataType extends DIABaseType and provides a base abstract type for a subset of types defined as part of the single value datatypes.
IntegerType	Describes a single integer value.
FloatType	Describes a single float value.
TokenType	Describes a single token value.
BooleanType	Describes a single Boolean value.

5.3.3 Vector datatypes

5.3.3.1 Introduction

This subclause specifies syntax and semantics of vector datatypes.

5.3.3.2 Vector datatypes syntax

```

<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Vector Data Types -->
<schema
  version="ISO/IEC 21000-7"
  id="VectorDatatypes.xsd"
  targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <import namespace="urn:mpeg:mpeg7:schema:2001"
    schemaLocation="mpeg7-udp-2003.xsd"/>
  <include schemaLocation="DIA.xsd"/>

  <!-- ##### -->
  <!-- Definition of Vector Data Types -->
  <!-- ##### -->

  <complexType name="VectorDataType" abstract="true">
    <complexContent>
      <extension base="dia:DIABaseType"/>
    </complexContent>
  </complexType>

  <!-- ##### -->
  <!-- Definition of IntegerVectorType -->
  <!-- ##### -->

  <complexType name="IntegerVectorType">
    <complexContent>
      <extension base="dia:VectorDataType">
        <sequence>
          <element name="Vector" type="mpeg7:integerVector"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- ##### -->
  <!-- Definition of FloatVectorType -->
  <!-- ##### -->

  <complexType name="FloatVectorType">
    <complexContent>
      <extension base="dia:VectorDataType">
        <sequence>
          <element name="Vector" type="mpeg7:floatVector"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- ##### -->
  <!-- Definition of TokenVectorType -->
  <!-- ##### -->

```

```
<complexType name="TokenVectorType">
  <complexContent>
    <extension base="dia:VectorDataType">
      <sequence>
        <element name="Vector">
          <simpleType>
            <restriction>
              <simpleType>
                <list itemType="dia:oneTokenType"/>
              </simpleType>
              <minLength value="1"/>
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<simpleType name="oneTokenType">
  <restriction base="token">
    <pattern value="\c+"/>
  </restriction>
</simpleType>

<!-- ##### -->
<!-- Definition of BooleanVectorType -->
<!-- ##### -->

<complexType name="BooleanVectorType">
  <complexContent>
    <extension base="dia:VectorDataType">
      <sequence>
        <element name="Vector">
          <simpleType>
            <list itemType="boolean"/>
          </simpleType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>

</schema>
```

5.3.3.3 Vector datatypes semantics

Semantics of the VectorDataType, IntegerVectorType, FloatVectorType, TokenVectorType, and BooleanVectorType:

Name	Definition
VectorDataType	VectorDataType extends DIABaseType and provides a base abstract type for a subset of types defined as part of the vector datatypes.
IntegerVectorType	Describes a vector of integer values.
FloatVectorType	Describes a vector of float values.
TokenVectorType	Describes a vector of token values.
BooleanVectorType	Describes a vector of Boolean values.

5.3.4 Matrix datatypes

5.3.4.1 Introduction

This subclause specifies syntax and semantics for matrix datatypes. Additionally, it includes the definition of the `NAType` which allows the inclusion of NA values within a matrix.

5.3.4.2 Matrix datatypes syntax

```
<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Matrix Data Types -->
<schema
  version="ISO/IEC 21000-7"
  id="MatrixDatatypes.xsd"
  targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <import namespace="urn:mpeg:mpeg7:schema:2001"
    schemaLocation="mpeg7-udp-2003.xsd"/>
  <include schemaLocation="DIA.xsd"/>

  <!-- ##### -->
  <!-- Definition of Matrix Data Types -->
  <!-- ##### -->

  <complexType name="MatrixDataType" abstract="true">
    <complexContent>
      <extension base="dia:DIABaseType">
        <attribute ref="mpeg7:dim" use="required"/>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="NAType">
    <restriction base="token">
      <enumeration value="NA"/>
    </restriction>
  </simpleType>

  <!-- ##### -->
  <!-- Definition of IntegerMatrixType -->
  <!-- ##### -->

  <complexType name="IntegerMatrixType">
    <complexContent>
      <extension base="dia:MatrixDataType">
        <sequence>
          <element name="Matrix" type="dia:NAIntegerMatrixType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="NAIntegerMatrixType">
    <list>
      <simpleType>
        <union memberTypes="dia:NAType integer"/>
      </simpleType>
    </list>
  </simpleType>
</schema>
```

```

        </simpleType>
    </list>
</simpleType>

<!-- ##### -->
<!-- Definition of FloatMatrixType -->
<!-- ##### -->

<complexType name="FloatMatrixType">
    <complexContent>
        <extension base="dia:MatrixDataType">
            <sequence>
                <element name="Matrix" type="dia:NAFloatMatrixType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<simpleType name="NAFloatMatrixType">
    <list>
        <simpleType>
            <union memberTypes="dia:NAType float"/>
        </simpleType>
    </list>
</simpleType>

<!-- ##### -->
<!-- Definition of TokenMatrixType -->
<!-- ##### -->

<complexType name="TokenMatrixType">
    <complexContent>
        <extension base="dia:MatrixDataType">
            <sequence>
                <element name="Matrix" type="dia:NATokenMatrixType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<simpleType name="NATokenMatrixType">
    <list>
        <simpleType>
            <union memberTypes="dia:NAType dia:aTokenType"/>
        </simpleType>
    </list>
</simpleType>
<simpleType name="aTokenType">
    <restriction base="token">
        <pattern value="\c+"/>
    </restriction>
</simpleType>

<!-- ##### -->
<!-- Definition of BooleanMatrixType -->
<!-- ##### -->

<complexType name="BooleanMatrixType">
    <complexContent>
        <extension base="dia:MatrixDataType">
            <sequence>
                <element name="Matrix" type="dia:NABooleanMatrixType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

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

  <simpleType name="NABooleanMatrixType">
    <list>
      <simpleType>
        <union memberTypes="dia:NAType boolean"/>
      </simpleType>
    </list>
  </simpleType>
</schema>

```

5.3.4.3 Matrix datatypes semantics

Semantics of the `MatrixDataType`, `IntegerMatrixType`, `FloatMatrixType`, `TokenMatrixType`, and `BooleanMatrixType`:

Name	Definition
<code>MatrixDataType</code>	<code>MatrixDataType</code> extends <code>DIABaseType</code> and provides a base abstract type for a subset of types defined as part of the matrix datatypes. The reserved word <code>NA</code> ("Not Available") can be specified to indicate that a certain value is invalid.
<code>IntegerMatrixType</code>	Describes a matrix of integer values.
<code>FloatMatrixType</code>	Describes a matrix of float values.
<code>TokenMatrixType</code>	Describes a matrix of token values.
<code>BooleanMatrixType</code>	Describes a matrix of Boolean values.

5.4 Base stack function

5.4.1 Introduction

This subclause specifies the syntax and semantics of the `BaseStackFunction` tool which describes a mathematical expression. The expression is represented by serializing it in terms of argument and operator elements. Argument elements are pushed onto a value stack. For operator elements, one or more arguments are popped out of the value stack based on the definition of the operator, the operation is performed, and the result is pushed back onto the value stack.

After all the elements of a serialized stack expression have been processed, only the final value of the expression remains on the value stack. An example of a mathematical expression represented as a stack function is provided in Figure 3. A list of stack function operators are provided in the `StackFunctionOperatorCS` in Annex A.

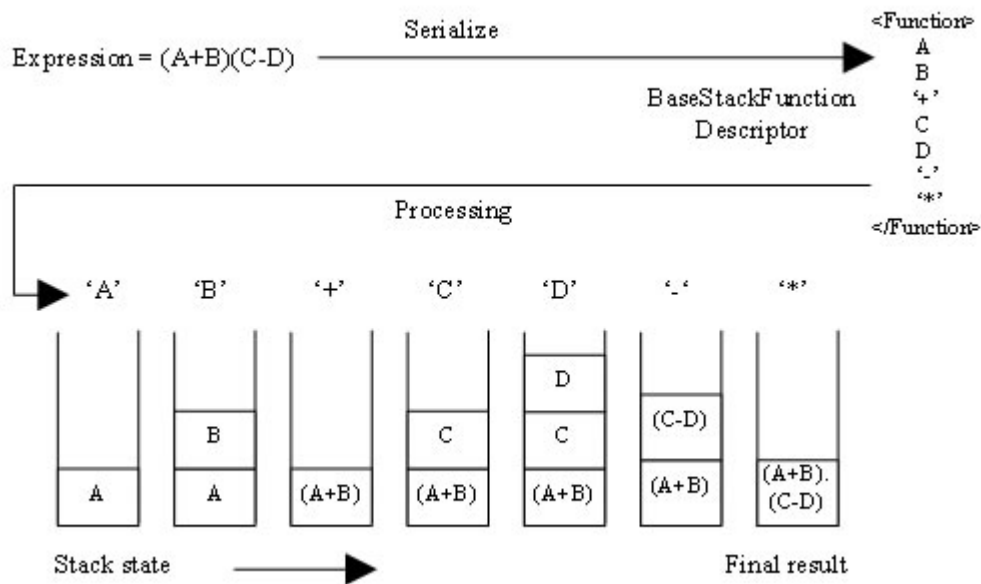


Figure 3 — Base stack function processing

5.4.2 Base stack function syntax

```
<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Stack Function Data Types -->
<schema
  version="ISO/IEC 21000-7"
  id="BaseStackFunction.xsd"
  targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <import namespace="urn:mpeg:mpeg7:schema:2001"
    schemaLocation="mpeg7-udp-2003.xsd"/>
  <include schemaLocation="DIA.xsd"/>

  <!-- ##### -->
  <!-- Definition of BaseStackFunctionType -->
  <!-- ##### -->

  <complexType name="BaseStackFunctionType">
    <complexContent>
      <extension base="dia:DIABaseType">
        <choice maxOccurs="unbounded">
          <element name="Argument" type="dia:ArgumentType"/>
          <element name="Operation">
            <complexType>
              <attribute name="operator"
                type="mpeg7:termReferenceType" use="required"/>
            </complexType>
          </element>
        </choice>
      </extension>
    </complexContent>
  </complexType>
  <!-- ##### -->
```

```

<!-- Definition of ArgumentType -->
<!-- ##### -->

<complexType name="ArgumentType" abstract="true">
  <complexContent>
    <extension base="dia:DIABaseType"/>
  </complexContent>
</complexType>

</schema>

```

5.4.3 Base stack function semantics

Semantics of the BaseStackFunctionType:

Name	Definition
BaseStackFunctionType	Tool for describing a mathematical expression. All the child elements of a stack function will be processed sequentially from top to bottom. The child element is pushed to the value stack if it is an <code>Argument</code> . If the child element is an <code>Operation</code> a corresponding number of values will be popped from the value stack and used as operands with the <code>operator</code> . The result value will then be pushed back to the value stack. After all the child elements of a stack expression are processed, only the final value of the stack expression should remain on the value stack.
Argument	Describes the input argument.
Operation	Describes the operation to be performed.
operator	Describes the operation. The CS that must be used for this purpose is the <code>StackFunctionOperatorCS</code> defined in annex A.2.15.

Implementations of the stack function need not implement any operators that are not listed in the CS. Implementations of the stack function should implement all operators in the CS. Stack function writers may utilize any operators that are not listed in the CS, but this will limit interoperability.

Semantics of the ArgumentType:

Name	Definition
ArgumentType	<code>ArgumentType</code> extends <code>DIABaseType</code> and provides a base abstract type for a subset of types defined as part of subclause 5.5.

5.5 Argument types and extensions

5.5.1 Introduction

This subclause specifies syntax and semantics of possible argument types and its extensions.

5.5.2 Argument types and extensions syntax

```

<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Stack Function Data Types -->
<schema
  version="ISO/IEC 21000-7"
  id="ArgumentTypes.xsd"
  targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <import namespace="urn:mpeg:mpeg7:schema:2001"
    schemaLocation="mpeg7-udp-2003.xsd"/>
  <include schemaLocation="BaseStackFunction.xsd"/>
  <include schemaLocation="SingleValueDatatypes.xsd"/>

  <!-- ##### -->
  <!-- Definition of VariableRefType -->
  <!-- ##### -->

  <complexType name="VariableRefType" abstract="true">
    <complexContent>
      <extension base="dia:ArgumentType">
        <attribute name="delay" type="nonNegativeInteger" use="optional"
          default="0"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- ##### -->
  <!-- Definition of SemanticalRefType -->
  <!-- ##### -->

  <complexType name="SemanticalRefType">
    <complexContent>
      <extension base="dia:VariableRefType">
        <attribute name="semantics" type="mpeg7:termReferenceType"
          use="required"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- ##### -->
  <!-- Definition InternalIOPinRefType -->
  <!-- ##### -->

  <complexType name="InternalIOPinRefType">
    <complexContent>
      <extension base="dia:VariableRefType">
        <attribute name="iOPinRef" type="IDREF" use="required"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- ##### -->
  <!-- Definition ExternalIOPinRefType -->
  <!-- ##### -->

```

```

<complexType name="ExternalIOPinRefType">
  <complexContent>
    <extension base="dia:VariableRefType">
      <attribute name="iOPinRef" type="anyURI" use="required"/>
    </extension>
  </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of GetValueType -->
<!-- ##### -->

<complexType name="GetValueType" abstract="true">
  <complexContent>
    <extension base="dia:ArgumentType"/>
  </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of ConstantDataType -->
<!-- ##### -->

<complexType name="ConstantDataType">
  <complexContent>
    <extension base="dia:GetValueType">
      <sequence>
        <element name="Constant" type="dia:ValueDataType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of ExternalDataRefType and derived types -->
<!-- ##### -->

<complexType name="ExternalDataRefType" abstract="true">
  <complexContent>
    <extension base="dia:GetValueType">
      <attribute name="uri" type="anyURI" use="required"/>
    </extension>
  </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of ExternalIntegerDataRefType -->
<!-- ##### -->

<complexType name="ExternalIntegerDataRefType">
  <complexContent>
    <extension base="dia:ExternalDataRefType"/>
  </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of ExternalFloatDataRefType -->
<!-- ##### -->

<complexType name="ExternalFloatDataRefType">
  <complexContent>
    <extension base="dia:ExternalDataRefType"/>
  </complexContent>
</complexType>

```

```
<!-- ##### -->
<!-- Definition of ExternalNMTokenDataRefType -->
<!-- ##### -->

<complexType name="ExternalNMTokenDataRefType">
  <complexContent>
    <extension base="dia:ExternalDataRefType"/>
  </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of ExternalBooleanDataRefType -->
<!-- ##### -->

<complexType name="ExternalBooleanDataRefType">
  <complexContent>
    <extension base="dia:ExternalDataRefType"/>
  </complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of SemanticalDataRefType -->
<!-- ##### -->

<complexType name="SemanticalDataRefType">
  <complexContent>
    <extension base="dia:GetValueType">
      <attribute name="semantics" type="mpeg7:termReferenceType"
        use="required"/>
      <attribute name="delay" type="nonNegativeInteger" use="optional"
        default="0"/>
    </extension>
  </complexContent>
</complexType>

</schema>
```

5.5.3 Argument types and extensions semantics

Semantics of the VariableRefType:

Name	Definition
VariableRefType	VariableRefType extends ArgumentType and provides a base abstract type for a subset of types defined as part of this subclause.
delay	Describes the delay value as a non-negative integer, in cases where the variable represents information from successive decision making units (frame, GOP, ROI etc.) in a resource, referred to as adaptation units. If delay is greater than zero, then the decision value for the variable delayed by that many adaptation units is used.

Semantics of the SemanticalRefType:

Name	Definition
SemanticalRefType	Describes a variable by semantical reference.

Name	Definition
semantics	Describes the semantics of the variable referenced. A CS that may be used for this purpose are the AdaptationQoS CS or MediaInformationCS defined in annex A.2.2 and A.2.10 respectively. Any CS term that refers to a Digital Item property without specifying whether it refers to the property of the input or output Digital Item is to be interpreted as referring to the property of the output Digital Item.

Semantics of the InternalIOPinRefType:

Name	Definition
InternalIOPinRefType	Describes an internal reference to an AdaptationQoS IOPin. Syntax and semantics of the AdaptationQoS IOPin are defined in clause 9.
iOPinRef	Describes the referenced IOPin.

Semantics of the ExternalIOPinRefType:

Name	Definition
ExternalIOPinRefType	Describes an external reference to an AdaptationQoS IOPin. Syntax and semantics of the AdaptationQoS IOPin are defined in clause 9.
iOPinRef	Describes the referenced IOPin.

Semantics of the GetValueType:

Name	Definition
GetValueType	GetValueType extends ArgumentType and provides a base abstract type for describing a value or a reference to a value.

Semantics of the ConstantDataType:

Name	Definition
ConstantDataType	Describes a constant.
Constant	Describes the value of the constant by use of type ValueDataType.

Semantics of the ExternalDataRefType:

Name	Definition
ExternalDataRefType	ExternalDataRefType extends GetValueType and provides a base abstract type for referencing external data by use of the XPointer framework.
uri	Describes the URI fragment identifier to be evaluated.

Semantics of the `ExternalIntegerDataRefType`:

<i>Name</i>	<i>Definition</i>
<code>ExternalIntegerDataRefType</code>	Describes an integer value which must be returned by the URI fragment identifier evaluation.

Semantics of the `ExternalFloatDataRefType`:

<i>Name</i>	<i>Definition</i>
<code>ExternalFloatDataRefType</code>	Describes an float value which must be returned by the URI fragment identifier evaluation.

Semantics of the `ExternalNMTokenDataRefType`:

<i>Name</i>	<i>Definition</i>
<code>ExternalNMTokenDataRefType</code>	Describes an <code>xsd:NMTOKEN</code> value which must be returned by the URI fragment identifier evaluation.

Semantics of the `ExternalBooleanDataRefType`:

<i>Name</i>	<i>Definition</i>
<code>ExternalBooleanDataRefType</code>	Describes an boolean value which must be returned by the URI fragment identifier evaluation.

Semantics of the `SemanticalDataRefType`:

<i>Name</i>	<i>Definition</i>
<code>SemanticalDataRefType</code>	Describes the value as a reference to its semantics.
<code>semantics</code>	Describes the semantics of the value referenced. A CS that may be used for this purpose are the <code>AdaptationQoS</code> , <code>MediaInformation</code> , or <code>SegmentDecompositionInfo</code> defined in annex A.2.2, A.2.10, and A.2.14 respectively. Any CS term that refers to a Digital Item property without specifying whether it refers to the property of the input or output Digital Item is to be interpreted as referring to the property of the input Digital Item.
<code>delay</code>	Describes the delay value as a non-negative integer, in cases where the semantics represents information from successive decision making units (frame, GOP, ROI etc.) in a resource, referred to as adaptation units. If delay is greater than zero, then the value for the semantics delayed by that many adaptation units is used.

Any CS term referenced by the semantics attribute which itself refer to CS terms specified in this part of ISO/IEC 21000 or other specifications must return the full URN of this CS term when used in conjunction with the `GetValueType`, i.e., no aliasing is allowed.

EXAMPLE The following example shows the fragment of an `AdaptationQoS` description containing an `IOPin` with a `GetValue` element.

```
<DIA>
  <DescriptionMetadata>
    <ClassificationSchemeAlias alias="AQoS:"
      href="urn:mpeg:mpeg21:2003:01-DIA-AdaptationQoSCS:NS"/>
  </DescriptionMetadata>
  <Description xsi:type="AdaptationQoSType">
    <Module xsi:type="LookUpTableType">
      <!-- description of chosen data representation -->
    </Module>
    <!-- other possible AdaptationQoS modules -->
    <IOPin id="VIDEO_DECODING_FORMAT">
      <GetValue semantics="AQoS:6.5.4.6"/>
    </IOPin>
    <!-- other possible IOPins -->
  </Description>
</DIA>
```

The `semantic` attribute refers to a CS term which defines the video decoding format of a terminal. The video decoding format is itself defined by means of a CS term within ISO/IEC 15983-5. In this case, possible returned values must contain the full URN of the video coding format CS term, e.g., "urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:3.3" (MPEG-4 Visual Advanced Simple Profile), and not something like ":VCF:3.3" where ":VCF:" is an alias for "urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001".

6 Usage environment description tools

6.1 Introduction

This clause specifies tools for describing the usage environment. The usage environment includes the description of User characteristics, terminal capabilities, network characteristics and natural environment characteristics. These various properties of the usage environment can be used for Digital Item Adaptation.

To distinguish description tools that are intended to describe multiple entities from those that are intended to describe a single entity, the plural form of that entity is used in the naming of the tool. For instance, the `TerminalsType` as defined in subclause 6.5.2 is a tool that specifies the characteristics of multiple terminals, whereas the characteristics of a single terminal are specified by the `TerminalType` as defined in subclause 6.5.3. Such a design of the schema allows the semantics associated with the respective elements of the tools to be less ambiguous, but still provides sufficient flexibility to describe multiple entities. Readers should take note of this naming convention since it occurs in a number of places throughout this clause.

6.2 Schema wrapper

The syntax of description tools specified in this subclause is provided as a collection of schema components, consisting notably in type definitions and element declarations. In order to form a valid schema document, these schema components should be gathered in a schema document with the following declaration defining in particular the target namespace, the namespaces prefixes and the imported namespaces.

```

<?xml version="1.0"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- Schema for Usage Environment Description Tools -->
<schema
  version="ISO/IEC 21000-7"
  id="UED.xsd"
  targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <import namespace="urn:mpeg:mpeg7:schema:2001"
    schemaLocation="mpeg7-udp-2003.xsd"/>
  <include schemaLocation="DIA.xsd"/>

```

Additionally, the following line should be appended to the resulting schema document in order to obtain a well-formed XML document.

```
</schema>
```

6.3 Usage environment

6.3.1 Introduction

This subclause specifies a container for descriptions of various usage environment properties. The various properties of the usage environment are derived from the abstract `UsageEnvironmentPropertyBaseType`.

6.3.2 Usage environment syntax

```

<!-- ##### -->
<!-- Definition of UsageEnvironment -->
<!-- ##### -->

<complexType name="UsageEnvironmentType">
  <complexContent>
    <extension base="dia:DIADescriptionType">
      <sequence>
        <element name="UsageEnvironmentProperty"
          type="dia:UsageEnvironmentPropertyBaseType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="UsageEnvironmentPropertyBaseType" abstract="true">
  <complexContent>
    <extension base="dia:DIADescriptionType"/>
  </complexContent>
</complexType>

```

6.3.3 Usage environment semantics

Semantics of the `UsageEnvironmentType`:

Name	Definition
<code>UsageEnvironmentType</code>	Tool for describing the usage environment.
<code>UsageEnvironmentProperty</code>	Describes a property of the usage environment, such as User characteristics, or terminal capabilities, or network characteristics, or natural environment characteristics.

Semantics of the `UsageEnvironmentPropertyBaseType`:

Name	Definition
<code>UsageEnvironmentPropertyBaseType</code>	<code>UsageEnvironmentPropertyBaseType</code> extends <code>DIADescriptionType</code> and provides a base abstract type for types that describe different properties of the usage environment.

6.4 User characteristics

6.4.1 Introduction

This subclause specifies tools for describing various characteristics of Users, including general User information, usage preferences and usage history, presentation preferences, accessibility characteristics, mobility characteristics and destination.

6.4.2 Users

6.4.2.1 Introduction

This subclause specifies the characteristics of multiple Users, where the characteristics of a single User are specified by the `User` element in the `UsersType`.

6.4.2.2 Users syntax

```
<!-- ##### -->
<!-- Definition of Users -->
<!-- ##### -->

<complexType name="UsersType">
  <complexContent>
    <extension base="dia:UsageEnvironmentPropertyBaseType">
      <sequence>
        <element name="User" type="dia:UserType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

6.4.2.3 Users semantics

Semantics of the `UsersType`:

Name	Definition
<code>UsersType</code>	Tool for describing the characteristics of Users.
<code>User</code>	Describes a single User in terms of general User information, content preferences, presentation preferences, accessibility characteristics, mobility characteristics and destination.

6.4.3 User

6.4.3.1 Introduction

This subclause specifies the characteristics of a single User. Specific characteristics include general User information, usage preferences and history, presentation preferences, accessibility characteristics, mobility characteristics and destination.

6.4.3.2 User syntax

```
<!-- ##### -->
<!-- Definition of User -->
<!-- ##### -->

<complexType name="UserType">
  <complexContent>
    <extension base="dia:DIABaseType">
      <sequence>
        <element name="UserCharacteristic"
          type="dia:UserCharacteristicBaseType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="UserCharacteristicBaseType" abstract="true">
  <complexContent>
    <extension base="dia:DIABaseType"/>
  </complexContent>
</complexType>
```

6.4.3.3 User semantics

Semantics of the `UserType`:

Name	Definition
<code>UserType</code>	Tool for describing characteristics of a single User.
<code>UserCharacteristic</code>	Describes a characteristic of the User, which may be related to general User information, content preferences, presentation preferences, accessibility characteristics, mobility characteristics and destination.

Semantics of the `UserCharacteristicBaseType`:

Name	Definition
<code>UserCharacteristicBaseType</code>	<code>UserCharacteristicBaseType</code> extends <code>DIADescriptionType</code> and provides a base abstract type for a subset of types defined as part of the <code>UserType</code> .

6.4.4 UserInfo

6.4.4.1 Introduction

This subclause specifies general information about Users, such as name and contact information, where a User can be a person, a group of persons, or an organization.

6.4.4.2 UserInfo syntax

```
<!-- ##### -->
<!-- Definition of UserInfo -->
<!-- ##### -->

<complexType name="UserInfoType">
  <complexContent>
    <extension base="dia:UserCharacteristicBaseType">
      <sequence>
        <element name="UserInfo" type="mpeg7:AgentType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

6.4.4.3 UserInfo semantics

Semantics of the `UserInfoType`:

Name	Definition
<code>UserInfoType</code>	Tool for describing general information about a User.
<code>UserInfo</code>	Describes general characteristics of a User such as name and contact information. A User can be a person, a group of persons, or an organization. The syntax and semantics of <code>AgentType</code> is specified in ISO/IEC 15938-5.

6.4.4.4 UserInfo example

EXAMPLE The following example shows the use of `UserInfo` expressing characteristics of a User named John Doe.

```
<DIA>
  <Description xsi:type="UsageEnvironmentType">
    <UsageEnvironmentProperty xsi:type="UsersType">
      <User>
        <UserCharacteristic xsi:type="UserInfoType">
          <UserInfo xsi:type="mpeg7:PersonType">
            <mpeg7:Name>
              <mpeg7:GivenName>John</mpeg7:GivenName>
              <mpeg7:FamilyName>Doe</mpeg7:FamilyName>
            </mpeg7:Name>
          </UserInfo>
        </UserCharacteristic>
      </User>
    </UsageEnvironmentProperty>
  </Description>
</DIA>
```

6.4.5 UsagePreferences

6.4.5.1 Introduction

This subclause specifies a tool for describing the usage preferences of a User related to the type and content of Digital Items. The `UsagePreferences` tool is essentially a container of various tools that directly describe the preferences of a User. Specifically, these include tools of preferences related to the creation of Digital Items (e.g., created when, where, by whom), preferences related to the classification of Digital Items (e.g., form, genre, languages), preferences related to the dissemination of Digital Items (e.g., format, location, disseminator), and preferences related to the type and content of summaries of Digital Items (e.g., duration of an audiovisual summary). The syntax and semantics of the `UserPreferencesType` is specified in Part 5 of ISO/IEC 15938.

6.4.5.2 UserPreferences syntax

```
<!-- ##### -->
<!-- Definition of UsagePreferences -->
<!-- ##### -->

<complexType name="UsagePreferencesType">
  <complexContent>
    <extension base="dia:UserCharacteristicBaseType">
      <sequence>
        <element name="UsagePreferences" type="mpeg7:UserPreferencesType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```