
**Information technology — Multimedia
service platform technologies —**

**Part 4:
Elementary services**

*Technologies de l'information — Technologies de la plate-forme de
services multimédia —*

Partie 4: Services élémentaires



Reference number
ISO/IEC 23006-4:2013(E)

AVA Documents
Document Sharing Hub
© ISO/IEC 2013

© ISO/IEC 2013

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
2 Normative references	1
3 Terms, definitions and abbreviated terms	3
3.1 Terms and definitions	3
3.2 Abbreviated terms	5
4 Namespaces and conventions	6
4.1 Namespaces.....	6
4.2 Namespace convention	9
4.3 Conventions	12
5 Elementary Services	13
5.1 Introduction.....	13
5.2 Base data types, elements, and common messages	25
5.3 MPEG-M service Schema wrapper	47
5.4 Session control and status polling.....	49
5.5 The Authenticate Services	56
5.6 The Authorize Services.....	65
5.7 The Check With Services.....	68
5.8 The Create Services	74
5.9 The Deliver Services	86
5.10 The Describe Services	94
5.11 The Identify Services.....	120
5.12 The Negotiate Services	135
5.13 The Package Services.....	159
5.14 The Post Services.....	164
5.15 The Present Services	166
5.16 The Process Services	171
5.17 The Request Services	181
5.18 The Revoke Services.....	202
5.19 The Search Services	210
5.20 The Store Services	226
5.21 The Transact Services	236
5.22 The Verify Services	259
Annex A (normative) Classification Schemes for IPTV offering discovery sections	269
A.1 General	269
A.2 Classification Scheme for ETSI IPTV (DVB-IP) offering discovery sections.....	269
A.3 Classification Scheme for ATIS IIF offering discovery sections.....	270
Annex B (normative) Service Types for Process Content.....	272
B.1 General	272
B.2 Recognize Speech.....	272
B.3 Synthesize Speech	275
B.4 Process Language.....	278
B.5 Translate Language.....	282
B.6 Extract Sensory Information	286
B.7 Content Adaptation	290
B.8 Resource Transcoding.....	295
B.9 Stream Repurposing	299

Annex C (normative) Schema for Service Instance Declaration	308
C.1 General.....	308
C.2 Schema Definition of Service Instance Declaration.....	308
C.3 Semantics of Service Instance Declaration	309
Annex D (normative) Usage of HTTP responses for response messages.....	311
D.1 General.....	311
Annex E (informative) Metadata Representation.....	312
E.1 General.....	312
E.2 Content metadata.....	312
E.3 Device Metadata.....	318
E.4 Service Instance Declaration Metadata	320
E.5 User Metadata	321
Annex F (normative) DIDL restriction for MPEG-M services	324
F.1 Purpose.....	324
F.2 MPEG-M DIDL Profile	324
Annex G (normative) Classification Schemes	339
G.1 General.....	339
G.2 AuctionModelCS	339
G.3 IssueTypeCS	343
G.4 TransferProtocolCS	344
G.5 TaggingTypeCS.....	345
G.6 RTPMediaTypeCS	346
G.7 ProtocolBindingCS.....	347
Annex H (normative) BPMN 2.0 XML representation of Elementary Services.....	348
H.1 General.....	348
H.2 OMG BPMN 2.0 Graphical Description	348
H.3 OMG BPMN 2.0 XML Description	351
Bibliography.....	358

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.

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

ISO/IEC 23006-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 23006-4:2010), which has been technically revised.

ISO/IEC 23006 consists of the following parts, under the general title *Information technology — Multimedia service platform technologies*:

- *Part 1: Architecture*
- *Part 2: MPEG extensible middleware (MXM) API*
- *Part 3: Conformance and reference software*
- *Part 4: Elementary services*
- *Part 5: Service aggregation*

Introduction

ISO/IEC 23006 is a suite of standards that has been developed for the purpose of enabling the easy design and implementation of media-handling value chains whose devices interoperate because they are all based on the same set of technologies accessible from the middleware.

ISO/IEC 23006 is referred as MPEG Extensible Middleware (MXM) in its first edition, and it specifies an architecture (Part 1), an API (Part 2), a reference software (Part 3) and a set of protocols which MXM Devices had to adhere (Part 4).

ISO/IEC 23006 is referred as Multimedia Service Platform Technologies (MSPT) in its second edition, and it conserves the architecture and design philosophy of the first edition, but stressing the Service Oriented Architecture character. It specifies also how to combine elementary services into aggregated services (Part 5).

This second edition has been specified to address the demand of service specification for an advanced IPTV terminal (AIT). The ISO/IEC 23006 suite of standards also aims at leveraging on advanced technologies to bring into IPTV services the buoyancy of new exciting initiatives – sometimes assembling millions of users in a fortnight – that pop up almost every day with new features such as open APIs and the possibility for third parties to provide applications to those APIs.

This enables the development of a global market of:

- MSPT applications that can run on MSPT devices, like Advanced IPTV Terminals (AITs), thanks to the existence of a standard MSPT application API
- MSPT devices executing MSPT applications thanks to the existence of a standard MSPT architecture
- MSPT engines thanks to the existence of standard MSPT architecture and standard APIs
- Innovative business models because of the ease to design and implement media-handling value chains whose devices interoperate because they are all based on the same set of technologies, especially MPEG technologies.

Information technology — Multimedia service platform technologies —

Part 4: Elementary services

1 Scope

This part of ISO/IEC 23006 specifies a set of Elementary Services and protocols enabling distributed applications to exchange information related to content items and parts thereof, including all the necessary Operations on MPEG-related Entities: Content, Contract, Device, Event, License, Service and User. These operations are defined to be the following: Authenticate, Authorize, Check With, Create, Deliver, Describe, Identify, Install, Interact With, Negotiate, Package, Post, Present, Process, Request, Revoke, Search, Store, Transact, Uninstall and Verify. Elementary Services can be combined in well defined sequences to build Aggregated Services, both of them being called in general Multimedia Services. ISO/IEC 23006 (all parts) will be referred to as MPEG-M for short in the text. The Multimedia Services are provided by and consumed by Multimedia Devices in a MSPT ecosystem, an example of which is the Advanced IPTV Terminal.

2 Normative references

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

ISO/IEC 14496-13, *Information technology — Coding of audio-visual objects — Part 13: Intellectual Property Management and Protection (IPMP) extensions*

ISO/IEC 15938-1, *Information technology — Multimedia content description interface — Part 1: Systems*

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

ISO/IEC 15938-12, *Information technology — Multimedia content description interface — Part 12: Query format*

ISO/IEC 21000-2, *Information technology — Multimedia framework (MPEG-21) — Part 2: Digital Item Declaration*

ISO/IEC 21000-3, *Information technology — Multimedia framework (MPEG-21) — Part 3: Digital Item Identification*

ISO/IEC 21000-4, *Information technology — Multimedia framework (MPEG-21) — Part 4: Intellectual Property Management and Protection Components*

ISO/IEC 21000-5, *Information technology — Multimedia framework (MPEG-21) — Part 5: Rights Expression Language*

ISO/IEC 21000-15, *Information technology — Multimedia framework (MPEG-21) — Part 15: Event Reporting*

ISO/IEC 21000-17, *Information technology — Multimedia framework (MPEG-21) — Part 17: Fragment Identification of MPEG Resources*

ISO/IEC 21000-20, *Information technology — Multimedia framework (MPEG-21) — Part 20: Contract Expression Language*

ISO/IEC 23000-5, *Information technology — Multimedia application format (MPEG-A) — Part 5: Media streaming application format*

ISO/IEC 23001-2, *Information technology — MPEG systems technologies — Part 2: Fragment request units*

ISO/IEC 23001-3, *Information technology — MPEG systems technologies — Part 3: XML IPMP messages*

ISO/IEC 23005-2, *Information technology — Media context and control — Part 2: Control Information*

ISO/IEC 23005-3, *Information technology — Media context and control — Part 3: Sensory Information*

ISO/IEC 23006-1, *Information technology — Multimedia service platform technologies — Part 1: Architecture*

ISO/IEC 23006-2, *Information technology — Multimedia service platform technologies — Part 2: MPEG Extensible Middleware (MXM) API*

ISO/IEC 23006-3, *Information technology — Multimedia service platform technologies — Part 3: Reference software*

IETF RFC 2616, *Hypertext Transfer Protocol — HTTP/1.1*, IETF Request For Comments, June 1999

IETF RFC 3614, *A Uniform Resource Name (URN) Namespace for the Motion Picture Experts Group (MPEG)*, IETF Request For Comments, September 2003

OASIS SAML-CORE-2.0-OS, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, OASIS Standard, 15 March 2005

OMG BPMN 2.0, *Business Process Model and Notation (BPMN) Version 2.0*, Object Management Group, January 2011

W3C SOAP, *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*, W3C Recommendation, 27 April 2007

W3C WSDL, *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*, W3C Recommendation, 26 June 2007

W3C XML, *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C Recommendation, 29 September 2006

W3C XMLNAMES, *Namespaces in XML 1.0 (Second Edition)*, W3C Recommendation, 16 August 2006

W3C XMLSCHEMA, *XML Schema Part 1: Structures Second Edition and XML Schema Part 2: Datatypes Second Edition*, W3C Recommendations, 28 October 2004

W3C XPATH1, *XML Path Language (XPath) Version 1.0*, W3C Recommendation, 16 November 1999

W3C XSL, *XSL Transformations (XSLT) Version 2.0*, W3C Recommendation, 23 January 2007

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1

Aggregated Service

service resulting from the combination of **Elementary Services**

3.1.2

Elementary Service

basic unit of **service**

3.1.3

content

Digital Item and its component elements, namely resources (e.g., media, scripts, executable), identifiers, descriptions (e.g., metadata), event reports

3.1.4

contract

set of **metadata**, **licenses**, promises and signers agreed by **Users** of a multimedia value chain, where a promise is a signed collection of statements about, e.g., obligations, prohibitions and assertions, and a signer is a **user** whose signature makes the contract valid

3.1.5

device

hardware/software or simply software apparatus that enables a **user** to play a role in multimedia **value chains**

3.1.6

event

performance of a specified set of functions or Operations

3.1.7

entity

one of the following elements in the multimedia value chain: **content**, **contract**, **device**, **event**, **license**, **service**, and **user**

3.1.8

governance

ability to control, direct or oversee the behavior of each **entity** or operation in an multimedia **value chain**

3.1.9

license

collection of authorizations, conditions and payment terms granted by a **user** to other **users**

3.1.10

operation

process in which an **Entity** is altered or manipulated

3.1.11

protocol

set of rules and data format used by two **Devices** to communicate

3.1.12

rate

function of expressing the perceived ranking of an **entity** related to a metric

3.1.13

remunerate

function of assigning money to a **user** in exchange of **rights**

3.1.14

reputation

measure of the credibility of or the possibility (e.g., legal) for a **user** to be a party in a transaction

3.1.15

resource

individually identifiable asset or a sequence of assets such as a video or audio clip, a 3D synthetic scene, an image, a textual asset, a 2D LAsER scene, a web page, a single program or a full 24 hour programming of a TV broadcast, a script or executable etc.

3.1.16

right

ability of a **user** to perform an Operation in the multimedia **value chain**

3.1.17

role

ability of a **user** to perform a set of Operations in the multimedia **value chain**

3.1.18

service

operation performed on an **entity** by a **user** on behalf of other **users**

3.1.19

service definition

specification of a **Service** in terms of interfaces, protocols as well as syntax and semantics of the protocol data formats

3.1.20

service instance

particular implementation of a **Service**

3.1.21

service instance declaration

description of a **Service Instance** in terms of provider, connection end-points, and usage conditions

3.1.22

service provider

user offering **services** to other **users**

3.1.23

space shifting

function of consuming content on a **device** (Device Shifting), or at a place (Space Shifting) or at a time (Time Shifting) different than the one intended by the rights holder

3.1.24

tag

free text descriptive information attached to an **entity**

3.1.25

user

any participant in multimedia **value chains**

3.1.26

value chain

collection of **users**, including Creators, End Users and Service Providers, that conform to this standard

3.2 Abbreviated terms

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

AIK	Attestation Identity Key
AIT	Advanced IPTV Terminal
AS	Aggregated Service
ATIS	Alliance for Telecommunications Industry Solutions
BBL	Bitstream Binding Language
BPMN	Business Process Model and Notation
CEL	Contract Expression Language
CRID	Content Reference Identifier
CS	Classification Scheme
DB	Database
DI	Digital Item
DIA	Digital Item Adaptation
DID	Digital Item Declaration
DIDL	Digital Item Declaration Language
DVB	Digital Video Broadcasting
DII	Digital Item Identification
EPG	Electronic Program Guide
ER	Event Report
ERR	Event Report Request
ES	Elementary Service
ETSI	European Telecommunications Standards Institute
FRU	Fragment Request Unit
FUU	Fragment Update Unit
HTTP	Hypertext Transfer Protocol
IIF	IPTV Interoperability Forum
IPMP	Intellectual Property Management and Protection
IPTV	Internet Protocol Television
LASeR	Lightweight Application Scene Representation
MDS	Multimedia Description Schemes
MIME	Multipurpose Internet Mail Extensions
MPEG	Moving Picture Experts Group
MPEG-4	Coding of audio-visual objects (see ISO/IEC 14496)
MPEG-7	Multimedia Content Description Interface Standard (see ISO/IEC 15938)
MPEG-21	Multimedia Framework (see ISO/IEC 21000)
MPEG-A	Multimedia application format (see ISO/IEC 23000)
MPEG-M	Multimedia Service Platform Technologies (see ISO/IEC 23006)
MPEG-V	Media Context and Control (see ISO/IEC 23005)

MPQF	MPEG Query Format
MSPT	Multimedia Service Platform Technologies
NER	Named Entity Recognition
OWL	Web Ontology Language
PCR	Platform Configuration Registers
POS	Part of Speech
RDF	Resource Description Framework
REL	Rights Expression Language
RTP	Real-Time Transport Protocol
SAML	Security Assertion Markup Language
SDP	Session Description Protocol
SID	Service Instance Declaration
SLA	Service Level Agreement
SNR	Signal-to-Noise Ratio
SP	Service Provider
SPARQL	SPARQL Protocol and RDF Query Language
SVC	Scalable Video Coding
TCPA	Trusted Computing Platform Alliance
TESP	Transact Entity Service Provider
TPM	Trusted Platform Module
TSS	Trusted Software Stack
UED	Usage Environment Description
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VAT	Value Added Tax
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformations

4 Namespaces and conventions

4.1 Namespaces

4.1.1 Introduction

URN namespaces defined in this Part of ISO/IEC 23006 conform to RFC 3614. The "standard name" used as prefix of the namespace specific string of URN namespaces defined in this Part of ISO/IEC 23006 is "mpegM".

NOTE Although RFC 3614 specifies the "standard name" to be case-insensitive, some tools might be incapable of handling this case insensitivity properly. Thus, alternative capitalizations are discouraged.

4.1.2 Schema namespaces

The value of the namespace URI for W3C XMLSCHEMA Schema definitions of base type used commonly by all Elementary Services in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:schema:01-base-NS:2012
```

The value of the namespace URI for XML Schema definitions of Elementary Service protocol messages in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:schema:02-service-NS:2012
```

The value of the namespace URI for XML Schema definitions of the BPMN 2.0 extension for message flow references in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:schema:04-bpmn-ext-mfr-NS:2012
```

The value of the namespace URI for XML Schema definitions of Service Instance Declarations in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:schema:05-sid-NS:2012
```

The value of the namespace URI for XML Schema definitions of the MPEG-21 DIDL extension in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:schema:06-didl-NS:2012
```

The value of the namespace URI for XML Schema definitions of the MPEG-M IPMP Info Profile in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012
```

The value of the namespace URI for XML Schema definitions of the MPEG-M IPMP DIDL Profile in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012
```

4.1.3 BPMN namespaces

The value of the namespace URI for BPMN 2.0 XML representations in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:bpmn:01-service-NS:2012
```

4.1.4 Service Type namespaces

The value of the namespace URI for XML Schema definitions of the Service Type "Recognize Speech" in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:service-type:01-recognize-speech-NS:2012
```

The value of the namespace URI for XML Schema definitions of the Service Type "Synthesize Speech" in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:service-type:02-synthesize-speech-NS:2012
```

The value of the namespace URI for XML Schema definitions of the Service Type "Extract Sensory Information" in this Part of ISO/IEC 23006 is:

`urn:mpeg:mpegM:service-type:03-extract-sensory-information-NS:2012`

The value of the namespace URI for XML Schema definitions of the Service Type "Content Adaptation" in this Part of ISO/IEC 23006 is:

`urn:mpeg:mpegM:service-type:04-content-adaptation-NS:2012`

The value of the namespace URI for XML Schema definitions of the Service Type "Process Language" in this Part of ISO/IEC 23006 is:

`urn:mpeg:mpegM:service-type:05-process-language-NS:2012`

The value of the namespace URI for XML Schema definitions of the Service Type "Translate Language" in this Part of ISO/IEC 23006 is:

`urn:mpeg:mpegM:service-type:05-translate-language-NS:2012`

The value of the namespace URI for XML Schema definitions of the Service Type "Resource Transcoding" in this Part of ISO/IEC 23006 is:

`urn:mpeg:mpegM:service-type:05-resource-transcoding-NS:2012`

The value of the namespace URI for XML Schema definitions of the Service Type "Stream Repurposing" in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:service-type:06-stream-repurposing-NS:2012
```

4.1.5 Classification Scheme namespaces

The value of the namespace URI for Classification Schemes for IPTV offering discovery sections in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:cs:02-iptvods-NS:2012
```

The value of the namespace URI for Classification Schemes for various MPEG-M Elementary Services in this Part of ISO/IEC 23006 is:

```
urn:mpeg:mpegM:cs:03-es-NS:2012
```

4.2 Namespace convention

For clarity, throughout this Part of ISO/IEC 23006, consistent namespace prefixes are used.

"xml:" and "xmlns:" are normative prefixes defined in W3C XMLNAMES. 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.

"xsi:" prefix is not normative. It is a naming convention in this document to refer to an element of the http://www.w3.org/2001/XMLSchema-instance namespace. All other prefixes used in either the text or examples of this specification are not normative, e.g., "mpegM:", "dia:".

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.

Unless specified otherwise, all unqualified descriptions fragments assume the default namespace "urn:mpeg:mpegM:schema:02-service-NS:2012".

In these descriptions fragments the different prefixes are bound to the namespaces as given in Table 1. The schema locations of the namespaces in Table 1 are only an informative indication as schema locations may change over time.

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

Prefix	Corresponding namespace	Schema location
xsd	http://www.w3.org/2001/XMLSchema	http://www.w3.org/2001/XMLSchema.xsd
xsi	http://www.w3.org/2001/XMLSchema-instance	
bbl	urn:mpeg:mpeg21:2007:01-BBL-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/dis/bbl.xsd
bpmn	http://www.omg.org/spec/BPMN/20100524/MODEL	http://www.omg.org/spec/BPMN/20100501/BPMN20.xsd

Prefix	Corresponding namespace	Schema location
bpmnext1	urn:mpeg:mpegM:schema:04-bpmn-ext-mfr-NS:2012	
ca	urn:mpeg:mpegM:service-type:04-content-adaptation-NS:2012	
cel	urn:mpeg:mpeg21:cel:core:2012	
cidl	urn:mpeg:mpeg-v:2010:01-CIDL-NS	
dc	http://purl.org/dc/elements/1.1/	http://dublincore.org/schemas/xmls/qdc/2008/02/11/dc.xsd
dcdv	urn:mpeg:mpeg-v:2010:01-DCDV-NS	
dia	urn:mpeg:mpeg21:2003:01-DIA-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/dia-2nd/UED-2nd.xsd
didl	urn:mpeg:mpeg21:2002:02-DIDL-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/did/didl.xsd
didl-msx	urn:mpeg:maf:schema:mediastreaming:DIDLextensions	Defined in ISO/IEC 23000-5:2011
didmodel	urn:mpeg:mpeg21:2002:02-DIDMODEL-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/did/didmodel.xsd
dii	urn:mpeg:mpeg21:2002:01-DII-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/dii/dii.xsd
dsig	http://www.w3.org/2000/09/xmlsig#	http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd
ebucore	urn:ebu:metadata-schema:ebuCore_2010	http://www.ebu.ch/metadata/schemas/EBUCore/20100820/EBU_CORE_20100820.xsd
erl	urn:mpeg:mpeg21:2005:01-ERL-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/er/er.xsd

Prefix	Corresponding namespace	Schema location
esi	urn:mpeg:mpegM:service-type:03-extract-sensory-information-NS:2012	
etsi	urn:dvb:metadata:iptv:sdns:2008-1	Defined in ETSI TS 102 034 [9]
fru	urn:mpeg:mpegB:schema:FragmentRequestUnits:2007	Defined in ISO/IEC 23001-2:2008
ipmpdidl	urn:mpeg:mpeg21:2004:01-IPMPDIDL-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/ipmp/ipmpdidl.xsd
ipmpinfo	urn:mpeg:mpeg21:2004:01-IPMPINFO-NS	standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/ipmp/ipmpgeneral.xsd
ipmpinfo-msx	urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007	Defined in ISO/IEC 23000-5:2011
ipmpmsg	urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007	Defined in ISO/IEC 23001-3:2008
mpeg4ipmp	urn:mpeg:mpeg4:IPMPSchema:2002	Defined in ISO/IEC 14496-13:2004
mpeg7	urn:mpeg:mpeg7:schema:2004	
mpqf	urn:mpeg:mpqf:schema:2008	Defined in ISO/IEC 15938-12
mpegm	urn:mpeg:mpegM:schema:02-service-NS:2012	
mpegmb	urn:mpeg:mpegM:schema:01-base-NS:2012	
mpegm-didl	urn:mpeg:mpegM:schema:06-didl-NS:2012	
mpegm-ipmpdidl	urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012	
mpegm-ipmpinfo	urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012	
rel-r	urn:mpeg:mpeg21:2003:01-REL-R-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/rel-r/rel-r.xsd
rel-sx	urn:mpeg:mpeg21:2003:01-REL-SX-NS	http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_files/rel-r/rel-sx.xsd
rs	urn:mpeg:mpegM:service-type:01-recognize-speech-NS:2012	

Prefix	Corresponding namespace	Schema location
saml	urn:oasis:names:tc:SAML:2.0:assertion	http://docs.oasis-open.org/security/saml/v2.0/saml-schema-assertion-2.0.xsd
samlp	urn:oasis:names:tc:SAML:2.0:protocol	http://docs.oasis-open.org/security/saml/v2.0/saml-schema-protocol-2.0.xsd
sedl	urn:mpeg:mpeg-v:2010:01-SEDL-NS	
sepv	urn:mpeg:mpeg-v:2010:01-SEPV-NS	
sev	urn:mpeg:mpeg-v:2010:01-SEV-NS	
sid	urn:mpeg:mpegM:schema:05-sid-NS:2012	
ss	urn:mpeg:mpegM:service-type:02-synthesize-speech-NS:2012	
tva	urn:tva:metadata:2010	Defined in ETSI TS 102 822-3-1 [7]
wsdl	http://www.w3.org/ns/wsdl	http://www.w3.org/2002/ws/desc/ns/wsdl20.xsd
xsl	http://www.w3.org/1999/XSL/Transform	http://www.w3.org/2007/schema-for-xslt20.xsd
xhtml	http://www.w3.org/1999/xhtml	http://www.w3.org/MarkUp/SCHEMA/xhtml11.xsd

NOTE In this Part of ISO/IEC 23006, it is discouraged to link to third-party XML Schema definitions in the `schemaLocation` attribute of `import` statements. Instead, local copies of these XML Schema definitions should be used. This measure increases availability of third-party XML Schema definitions and avoids excessive traffic [11].

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.

4.3 Conventions

Fixed-width font is used to indicate literal machine-readable character sequences.

The names of XML Schema attributes appear in fixed-width font in mixed case with an initial lower case letter. The names of XML Schema elements and types appear in fixed-width font in mixed case with an initial upper case letter.

XML Schema definitions are represented with orange amber background color and a double line black border.

XML examples are represented with gray background color and a single line black border.

XML Infosets of MPEG-7 Classification Schemes, XML declarations of Service Types, and BPMN 2.0 XML notations are represented with gray background color and a double line black border.

The protocol specifications for Elementary Services in this Part of ISO/IEC 23006 are represented by protocol tables. The first row of a *protocol table* is the header, which specifies the entities involved in this protocol. Each subsequent row describes a single step of the protocol run. For each step, the first column indicates the

step number, subsequent columns describe the action taken by the corresponding entity. If a step is not attributed to a single entity, then cells in that row are merged accordingly. Steps that are only executed depending on some conditions are marked with the keyword "(optional)" and are represented in light gray background color. The example below illustrates of such a protocol table.

Steps	Entity A	Entity B
1.	Entity A and Entity B perform some common action.	
2.	Entity A sends a message to Entity B.	
3.		Entity B processes the received message.
4.		(optional) Depending on the result of the processing, Entity B may send a response to Entity A.

5 Elementary Services

5.1 Introduction

5.1.1 Overview of Elementary Services

This subclause introduces the concept of Elementary Services, defines relevant terms, and provides guidelines and rules for the specification of Elementary Services. 0 gives an overview of Operations and Entities, as well as the corresponding Elementary Services. 5.1.3 to 5.1.8 provide guidelines and rules for the specification of Elementary Services and the specification of Service Instance Declarations.

An Elementary Service (e.g., *Search Content*) corresponds to an Operation (e.g., *Search*) that is performed for a kind of Entity (e.g., *Content*).

An Elementary Service (ES) is a Service of atomic nature, which cannot be usefully divided into smaller parts and does not normatively mandate other Services to be executed during its workflow.

The terms defined in Table 2 allow for generalizing statements about Elementary Services.

Table 2 — Definition of kinds of Elementary Services

Term	Definition	Example (informative)
<i>abstract Elementary Service</i>	Abstraction of all Elementary Services that correspond to a specific Operation, regardless of their kind of Entity. NOTE An <i>abstract Elementary Service</i> specifies the protocol and workflow that all regular Elementary Services derived from it share (unless specified otherwise).	<i>Search Entity</i>

Term	Definition	Example (informative)
<i>regular Elementary Service</i>	Corresponds to a specific Operation that is performed for a specific kind of Entity. The terms <i>regular Elementary Service</i> and <i>Elementary Service</i> are used synonymously. Elementary Services can be further classified by their complexity: — An Elementary Service with a simple request-response protocol is called <i>atomic Elementary Service</i>. — An Elementary Service with a protocol that uses more than these two messages is called <i>compound Elementary Service</i>. NOTE A <i>regular Elementary Service</i> may be derived from an abstract Elementary Service. NOTE A <i>regular Elementary Service</i> specifies the protocol, workflow, as well as the syntax and semantics of protocol data formats.	<i>Deliver Content</i> (a compound regular Elementary Service not derived from an abstract Elementary Service). <i>Search Content</i> (an atomic regular Elementary Service derived from <i>Search Entity</i>). <i>Authenticate Content</i> (an atomic regular Elementary Service specified according to, but not derived by, <i>Authenticate Entity</i> abstract Elementary Service)
<i>generic Elementary Service</i>	Elementary Service that requires a specific Service Type in order to be instantiated. NOTE A <i>generic Elementary Service</i> corresponds to a specific kind Entity but leaves the Operation generic.	<i>Process Content</i> (the only generic Elementary Service specified in this Part of ISO/IEC 23006).
<i>specific Elementary Service</i>	<i>Generic Elementary Service</i> that is associated with a Service Type. NOTE The Service Type further specifies the Operation that corresponds to the <i>specific Elementary Service</i>.	<i>Process Content</i> with Service Type "Recognize Speech".

5.1.2 Overview of Operations and Entities

The Entities defined in this Part of ISO/IEC 23006 are listed in Table 3.

Table 3 — Definition of Entities

Entity	Definition	Representation
Content	An ISO/IEC 21000 Digital Item and its component elements, namely Resources (e.g., media, scripts, executable), Identifiers, Descriptions (e.g., metadata) and Event Reports.	Specified in ISO/IEC 21000-2. An alternative representation is specified in Annex F.
Contract	A set of metadata, Licenses, promises and signers agreed by Users of an MPEG-M value chain, where a <i>promise</i> is a signed collection of statements about (e.g., obligations, prohibitions and assertions) and a <i>signer</i> is a User whose signature makes the Contract valid.	Specified in ISO/IEC 21000-20.
Device	A hardware/software or simply software apparatus that enables a User to play a role in MPEG-M value chains.	Specified in ISO/IEC 23000-5.
Event	The performance of a specified set of functions or Operations.	Specified in ISO/IEC 21000-15.
License	A collection of authorizations, conditions and payment terms granted by a User to other Users.	Specified in ISO/IEC 21000-5.
Service	A set of Operations performed by a User on behalf of other Users. A Service Instance is a particular implementation of a Service, typically described in terms of provider, connection end-points, and usage conditions.	The representation of Service Instances is specified in Annex C.
User	Any participant in MPEG-M value chains.	Specified in ISO/IEC 15938-5.

The Operations defined in this Part of ISO/IEC 23006 are listed in Table 4.

Table 4 — Definition of Operations

Operation	Definition
Authenticate	To confirm the identity of an Entity.
Authorize	To grant rights to perform certain operations.
Check With	To check if the User is allowed to act on the content based on a Contract or a License.
Create	To generate a data structure representing a Content, a Contract, or a License.
Deliver	To ask a Service Provider to transfer a Content or Contract from a Device or User to another Device or User.
Describe	To add descriptive information to an Entity.
Identify	To assign an identifier to an Entity.

Operation	Definition
Negotiate	To assist Users to seek agreement on a Transaction.
Package	To make a Content Item suitable for Delivery.
Post	To make a Content discoverable by other Entities.
Present	To make Contract or License available for consumption by the intended user.
Process	To make change to an Entity or combine Entities into another Entity.
Request	To call for the provision of an Entity.
Revoke	To discontinue the validity of an Entity.
Search	To find the URI or physical location of Entities satisfying given conditions.
Store	To transfer a digital Entity to a Service Provider for storage.
Transact	To grants Rights to an Entity in exchange of a remuneration.
Verify	To check the validity of an Entity.

Table 5 shows the Elementary Services for the combinations of Operations and Entities are defined in this Part of ISO/IEC 23006. The entries in the table refer to the subclause in which the respective Elementary Service is defined. Note that the column labeled "Entity" refers to abstract Elementary Services, from which the regular Elementary Services in that row are derived.

Table 5 — Elementary Services classified by Operations and Entities

	Entity	Content	Contract	Device	Event	License	Service	User
Authenticate	5.5.1	5.5.2	5.5.3			5.5.4		5.5.5
Authorize								5.6.1
Check With			5.7.1			5.7.2		
Create		5.8.1	5.8.2			5.8.3		
Deliver		5.9.1	5.9.2					
Describe	5.10.1	5.10.2		5.10.3			5.10.4	5.10.5
Identify	5.11.1	5.11.2	5.11.3	5.11.4		5.11.5		0
Negotiate	5.12.1		5.12.2			5.12.3		
Package		5.13.1						
Post		5.14.1						
Present	5.15.1		5.15.2			5.15.3		

	Entity	Content	Contract	Device	Event	License	Service	User
Process		5.16.1				5.16.2		
Request		5.17.1	5.17.2	5.17.3	5.17.4	5.17.5		
Revoke	5.18.1	5.18.2	5.18.3			5.18.4		
Search	5.19.1	5.19.2	5.19.3	5.19.4		5.19.5	5.19.6	5.19.7
Store		5.20.1	5.20.2		5.20.3	5.20.4		
Transact	5.21.1	5.21.2				5.21.3		
Verify			5.22.1	5.22.2		5.22.3		

5.1.3 General Service Definition

The following subclauses describe the relation between the *General Service Definition*, *Service Type*, and *Service Instance Declaration*.

A *General Service Definition* denotes the specification of interfaces, protocol specification as well as syntax and semantics of the protocol data formats of a Service. A General Service Definition can either specify an Elementary Service or an Aggregated Services. Examples of General Service Definitions are the Elementary Services specified in 5.5 to 5.22 of this part of ISO/IEC 23006 (called MPEG-M for short) and the Aggregated Services provided in 5.3 of ISO/IEC 23006-5. The terms *General Service Definition* and *Service Definition* are used synonymously throughout this standard.

A General Service Definition must comprise at least all of the following parts:

- Narrative description of the actions that the Service performs;
- BPMN 2.0 XML representation of a Collaboration and a Process for the Service Provider as a formal description of Service's workflow;
- Syntax of input and output messages, represented as XML Schema;
- Descriptions of semantics of input and output messages and of their parameters;
- Extension to the Service Instance Declaration, containing syntax and semantics of any additional parameters that should be published by a Service Provider.

The above defined parts are mandatory for the description of both Elementary Services and Aggregated Services.

This standard does not specify the representational format for the narrative description of actions, description of semantics of messages, and description of semantics of extensions to the SID. It is, however, recommended to follow the style of the specification of Elementary Services provided in this Part of ISO/IEC 23006.

The BPMN 2.0 Process and Collaboration instances must at least model the exchange of messages between the caller and the SP as defined in 5.1.8. For the description of Elementary Services, the BPMN Process should be of type BPMN Public Process. The BPMN messages should be mapped to XML data types of input and out messages.

NOTE The BPMN 2.0 Process and Collaboration instances of all Elementary Services specified in this part of ISO/IEC 23006 are collectively provided in Annex H. Since the protocols for most of these Elementary Services only consist of a request message and a corresponding response message, the BPMN representations are only included directly in the Service definitions where it is considered appropriate.

For the definition of Aggregated Services, it is not mandatory to describe the interaction with the Services of which the AS is composed. Optionally, this information can be provided in the definition, e.g., in the BPMN Process.

The BPMN Process instance must be represented as XML serialization. Optionally, an informative BPMN diagram of the BPMN Process instance can be included in the description of a Service. If a diagram is present, it must conform to the XML serialization of the BPMN Process.

For regular Services, the syntax definition of input and output messages comprises the definition of XML elements and any XML complex types necessary for instantiating these elements.

For abstract Services, the syntax definition of input and output messages shall only comprise the XML complex types for the messages, but not the message XML elements themselves. For regular Services based on abstract Services, the syntax definition of input and output messages shall comprise the message XML elements based on the XML complex types of the corresponding abstract Service.

EXAMPLE The Search Entity abstract Elementary Service defines the complex types `mpegm:SearchEntityRequestType` and `mpegm:SearchEntityResponseType`. The Search Content regular Elementary Service, which is based on the abstract ES, defines a message element `mpegm:SearchContentRequest` of type `mpegm:SearchEntityRequestType` and a message element `mpegm:SearchContentResponse` of type `mpegm:SearchEntityResponseType`.

The extension to the SID should be represented as an XML complex type that is derived from the `sid:ServiceInstanceDeclarationType` complex type.

Optionally, the description of a Service can contain examples about the usage of the Service.

The description of Services may contain extensions that are not specified in this International Standard.

5.1.4 Service Type

The Process Content elementary service defined in 5.16.1 provides a generic workflow for the processing of content. In order to specify the concrete task of this processing (e.g., Recognize Speech, Transcoding, etc.), 5.16.1 defines the concept of Service Types. Process Content is called a generic Elementary Service since it needs a Service Type in order to be implemented.

The Service Type of a Service Instance of Process Content must be provided in the Service Instance Declaration.

A non-exhaustive list of Service Types for Process Content is specified in Annex B.

Further examples of Service Types are provided in Annex A of ISO/IEC 23006-5. Note that ISO/IEC 23006-5 also specifies a mechanism for the representation of Service Types at the Service aggregation level.

Other parties may specify additional Service Types. This is done by making the `ServiceType` element, representing the formal specification of the Service Type, available. If the `ServiceType` element contains a `RequestParametersType` element, this specified extension to the `RequestParametersBaseType` must also be made available. If the `ServiceType` element contains a `CompletionParametersType` element, this specified extension to the `CompletionParametersBaseType` must also be made available. The XML Schema definition of these extensions shall contain a description of the purpose of the Service Type and the specification of the semantics of the extension represented as `xsd:annotation/xsd:documentation` elements.

5.1.5 Service Instance Declaration

A *Service Instance Declaration* describes a concrete realization of a Service where the Service is specified through a General Service Definition. A Service Instance Declaration is created by a Service Provider and specifies the concrete parameters that enable a client to invoke that Service Instance. A Service Instance Declaration must identify the General Service Definition (or Service Type) that it implements. Conversely, a General Service Definition (or a Service Type) can have multiple Service Instance Declarations implementing it.

A Service Instance Declaration comprises the following components:

- a) Identifier of the Service Instance
- b) Name of the Service Instance
- c) Identifier of the corresponding General Service Definition
- d) Indication of supported formats for metadata representation
- e) Provider (i.e., address, contact information, self-description, etc.)
- f) Connection end-point (i.e., where the client has to connect to in order to use the Service), e.g., IP address
- g) Protocol binding description that specifies how MPEG-M protocol messages are exchanged
- h) License and/or Contract that regulates the use of the Service Instance (optional)
- i) Description of the Service Instance (optional)
- j) Service availability period (optional)
- k) Application-specific data, e.g., further configuration (optional)
- l) Signature (optional).

If the signature is present, it always signs the entire Service Instance Declaration, including any extensions.

The preferred XML representation of Service Instance Declarations is specified in Annex C.

5.1.6 Message transport

This document does not specify the transport protocol over which the payload is delivered. The transport protocol must provide ordered message delivery. The transport protocol of a Service Instance is indicated by the SID as described in 5.1.5 and Annex C.

The communication for MPEG-M Services relies on mutual authentication between the involved entities as described in 5.2.5.1.

Response messages may be represented by HTTP responses in some cases as specified in Annex D.

5.1.7 Validation

Validating a document against the MPEG-M schema (as specified in W3C XMLSCHEMA) is necessary, but not sufficient, to determine its validity with respect to MPEG-M. After a document is validated against the MPEG-M schema, it shall also be subjected to additional validation rules. These additional rules are given below in the descriptions of the elements to which they pertain.

5.1.8 BPMN 2.0 XML representation of Service workflows

5.1.8.1 Introduction

OMG BPMN 2.0 specifies a notation to represent processes which is used in throughout this document to provide a formal representation Service workflows. BPMN 2.0 specifies both, a graphical notation and an XML representation for processes. The workflow of an Elementary Service is normatively specified through the XML representation of BPMN 2.0 Collaboration and Process instances.

Unless specified otherwise (see 5.1.8.2), the BPMN 2.0 XML representation of an Elementary Service must contain all of the following elements:

- a) A `bpmn:collaboration` that identifies the participants (i.e., client and SP) and the message flows between them.
- b) A `bpmn:process` representing a BPMN Public Process for the SP's workflow.
- c) `bpmn:message` elements for all messages exchanged between client and SP. Each `bpmn:message` references a `bpmn:itemDefinition`.
- d) `bpmn:itemDefinition` elements that reference the actual XML messages of the protocol data format.

The following rules apply to the `bpmn:collaboration` element:

The `bpmn:collaboration` shall specify the `id` attribute in order to allow references from the `bpmn:process` of the SP. The `name` attribute shall be the name of the Elementary Service (e.g., "Deliver Content").

The `bpmn:participant` elements shall specify the `name` attributes (e.g., "Client" for the client and "Deliver Content SP" for the SP). The `bpmn:participant` representing the SP must reference the `bpmn:process` of the Elementary Service through the `processRef` attribute. The `bpmn:participant` representing the client shall specify the `id` attribute in order to enable Service aggregation as explained in ISO/IEC 23006-5.

The `bpmn:messageFlow` elements must reference through their `sourceRef` and `targetRef` attributes the `bpmn:process` for the client and the BPMN Flow Nodes (i.e., Events, Tasks, Gateways) within the `bpmn:process` for the SP. Each `bpmn:messageFlow` element shall specify the `id` attribute in order to allow Aggregated Services to match message flows of the Services they are composed of (see 6.2 of ISO/IEC 23006-5). Unless specified otherwise (see 5.1.8.2), the `bpmn:messageFlow` elements must reference the corresponding `bpmn:messages` through the `messageRef` attribute.

The following rules apply to the `bpmn:process` element:

The `bpmn:process` must specify the `id` attribute in order to allow references from the `bpmn:participant` of the `bpmn:collaboration` and to enable Service aggregation as explained in ISO/IEC 23006-5. The `processType` attribute shall be set to "Public". The `bpmn:collaboration` of the Elementary Service shall be referenced through the `definitionalCollaborationRef` attribute.

The `bpmn:process` shall contain a `bpmn:startEvent`, which is triggered by the first message flow from the client to the SP. The `bpmn:startEvent` shall contain a `bpmn:messageEventDefinition` child element. Further BPMN Flow Nodes (i.e., Events, Tasks, Gateways, SubProcesses), and the `bpmn:sequenceFlow` elements connecting them, specify a high-level workflow for the SP. The Flow Nodes of the `bpmn:process` shall not reference any BPMN messages themselves; the connection is established by the `bpmn:messageFlow` elements of the `bpmn:collaboration`. Along the normal workflow, outgoing response messages should indicate successful execution of the respective tasks (i.e., they should contain an element derived from `ProtocolSuccessType`). Exception sequence flows should originate from BPMN Error Boundary Events and the response messages originating from BPMN Send Tasks along those sequence flows shall indicate failures (i.e., they should contain an element derived from `ProtocolFailureType`).

The following rules apply to the `bpmn:message` elements:

The `bpmn:message` must specify the `id` attribute in order to allow references from the `bpmn:messageFlow`. The `name` attribute shall be the name of the protocol message of the Elementary Service (e.g., "Deliver Content Request"). The corresponding `bpmn:itemDefinition` must be referenced through the `itemRef` attribute.

The following rules apply to the `bpmn:itemDefinition` elements:

The `bpmn:itemDefinition` must specify the `id` attribute in order to allow references from the `bpmn:message` elements. The XML element of the actual protocol message of the Elementary Service (e.g., `mpegm:DeliverContentRequest`) must be referenced through the `structureRef` attribute.

The representation of workflows of the Elementary Services defined in this document is provided in Annex H.

5.1.8.2 BPMN 2.0 XML representation of abstract base Elementary Service workflows

The binding of the workflow of abstract base Elementary Services to the derived actual Elementary Service is accomplished through the extensibility mechanism of BPMN 2.0.

In BPMN 2.0, a `bpmn:messageFlow` element represents the connection between two BPMN 2.0 Interaction Nodes (i.e., Pool/Participant, Activity, or Event), along which a message is sent. BPMN 2.0 envisions that the `bpmn:messageFlow` element specifies the reference to a `bpmn:message` element which links to the actual data type of the message. This mechanism is not sufficient for defining a generic workflow (e.g., for an abstract base Elementary Service) and concretizing it with data formats of messages (e.g., to obtain an actual Elementary Service).

The BPMN 2.0 extension for message flow references provides the `bpmn:message` element with an additional `bpmnext1:MessageFlowRef` child element in order to bind it to a `bpmn:messageFlow` element.

The XML Schema definition of the BPMN 2.0 extension for message flow references is given below.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:mpeg:mpegM:schema:04-bpmn-ext-mfr-NS:2012"
xmlns:bpmnext1="urn:mpeg:mpegM:schema:04-bpmn-ext-mfr-NS:2012"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <element name="MessageFlowRef" type="QName"/>
</schema>
```

Semantics of the `bpmnext1:MessageFlowRef`:

Name	Definition
<code>bpmnext1:MessageFlowRef</code>	Provides an association between <code>bpmn:messageFlow</code> and <code>bpmn:message</code>, similar to the <code>messageRef</code> attribute of the <code>bpmn:messageFlow</code>, but specifying it from the other end at the <code>bpmn:message</code> element. The <code>bpmnext1:MessageFlowRef</code> element shall only be used as an extension to <code>bpmn:message</code> elements. The value is a reference to a <code>bpmn:messageFlow</code> element.

If the Elementary Service has an abstract base Elementary Service (e.g., Search Entity is the abstract base Elementary Service for Search Content), then the following additional rules apply:

The surrounding bpmn:definitions root element must specify a bpmn:extension element. The mustUnderstand attribute of the bpmn:extension element must be set to "true". The definition attribute must reference the bpmnext1:MessageFlowRef element.

The BPMN 2.0 XML representation of the abstract base Elementary Service (e.g., Search Entity) must contain the bpmn:collaboration and bpmn:process elements as specified in 5.1.8. However, the bpmn:messageFlow elements must not reference any bpmn:message elements. Each bpmn:messageFlow element must specify the id attribute in order to be referenced by a bpmnext1:MessageFlowRef element in the BPMN representation of the actual Elementary Service (e.g., Search Content).

The BPMN 2.0 XML representation of the abstract base Elementary Service (e.g., Search Entity) must not contain any bpmn:message or bpmn:itemDefinition elements.

The BPMN 2.0 XML representation of the actual Elementary Service derived from the abstract base Elementary Service (e.g., Search Content) must contain the bpmn:message elements and bpmn:itemDefinition elements as specified in 5.1.8. Each bpmn:message element must contain a bpmn:extensionElements element which contains one or more bpmnext1:MessageFlowRef elements. Each bpmnext1:MessageFlowRef element must reference an individual corresponding bpmn:messageFlow element (e.g., the message flow for Search Entity Request) from the bpmn:collaboration of the corresponding abstract base Elementary Service.

The BPMN 2.0 XML representation of the actual Elementary Service (e.g., Search Content) must not contain any bpmn:collaboration or bpmn:process elements.

Examples for the BPMN 2.0 XML representation of abstract base Elementary Services are Describe Entity and Search Entity as provided in Annex H.

5.1.8.3 Examples

The example below shows the BPMN 2.0 XML representation for the Elementary Service Create Content (see 5.8.1).

```
<bpmn:definitions targetNamespace="urn:mpeg:mpegM:bpmn:01-service-NS:2012">
  <!-- ***** -->
  <!-- Collaboration, Process, and Messages for Create Content -->
  <!-- ***** -->
  <!-- NOTE: This is an atomic workflow. -->
  <bpmn:collaboration id="create_content_collaboration" name="Create Content">
    <bpmn:participant id="create_content_client" name="Client"/>
    <bpmn:participant name="Create Content SP" processRef="create_content_sp"/>
    <bpmn:messageFlow id="create_content_request_message_flow" name="Create
Content Request MessageFlow" sourceRef="create_content_client"
targetRef="create_content_start" messageRef="create_content_request"/>
    <bpmn:messageFlow id="create_content_reject_message_flow" name="Create
Content Reject MessageFlow" sourceRef="create_content_reject_request"
targetRef="create_content_client" messageRef="create_content_response"/>
    <bpmn:messageFlow id="create_content_response_message_flow" name="Create
Content Response MessageFlow" sourceRef="create_content_respond"
targetRef="create_content_client" messageRef="create_content_response"/>
  </bpmn:collaboration>
  <bpmn:process id="create_content_sp" processType="Public"
definitionalCollaborationRef="create_content_collaboration">
    <bpmn:startEvent id="create_content_start">
      <bpmn:messageEventDefinition/>
    </bpmn:startEvent>
```

```

    <bpmn:sequenceFlow sourceRef="create_content_start"
targetRef="create_content_fulfill_request"/>
    <bpmn:task id="create_content_fulfill_request" name="Fulfill Request"/>
    <bpmn:boundaryEvent id="create_content_fulfill_error_occurrence"
attachedToRef="create_content_fulfill_request">
    <bpmn:eventDefinition xsi:type="bpmn:tErrorEventDefinition"/>
    </bpmn:boundaryEvent>

    <bpmn:sequenceFlow sourceRef="create_content_fulfill_error_occurrence"
targetRef="create_content_reject_request"/>
    <bpmn:sendTask id="create_content_reject_request" name="Reject Request"/>
    <bpmn:sequenceFlow sourceRef="create_content_reject_request"
targetRef="create_content_error_end"/>
    <bpmn:endEvent id="create_content_error_end" name="Error">
    <bpmn:eventDefinition xsi:type="bpmn:tErrorEventDefinition"/>
    </bpmn:endEvent>

    <bpmn:sequenceFlow sourceRef="create_content_fulfill_request"
targetRef="create_content_respond"/>
    <bpmn:sendTask id="create_content_respond" name="Respond"/>
    <bpmn:sequenceFlow sourceRef="create_content_respond"
targetRef="create_content_success_end"/>
    <bpmn:endEvent id="create_content_success_end" name="Success"/>
    </bpmn:process>

    <bpmn:message id="create_content_request" name="CreateContentRequest"
itemRef="create_content_request_itemdef"/>
    <bpmn:message id="create_content_response" name="CreateContentResponse"
itemRef="create_content_response_itemdef"/>

    <bpmn:itemDefinition id="create_content_request_itemdef"
structureRef="mpegm:CreateContentRequest"/><!-- Binding of the bpmn:message to
the actual MPEG-M request message definition. -->
    <bpmn:itemDefinition id="create_content_response_itemdef"
structureRef="mpegm:CreateContentResponse"/><!-- Binding of the bpmn:message to
the actual MPEG-M response message definition. -->
    </bpmn:definitions>

```

The example below shows the BPMN 2.0 XML representations of the abstract base Elementary Service Search Entity (see 5.19.1) and of the actual Elementary Service Search Content (see 5.19.2) based on it.

```

<bpmn:definitions targetNamespace="urn:mpeg:mpegM:bpmn:01-service-NS:2012">

    <bpmn:extension definition="bpmnext1:MessageFlowRef" mustUnderstand="true">
    <bpmn:documentation>Used in bpmn:message to reference a
bpmn:messageFlow.</bpmn:documentation>
    </bpmn:extension>
    <!-- ***** -->
    <!-- Collaboration and Process for Search Entity -->
    <!-- ***** -->
    <!-- NOTE: This is an atomic workflow for an abstract base Elementary Service.
-->
    <bpmn:collaboration id="search_entity_collaboration" name="Search Entity">
    <bpmn:participant id="search_entity_client" name="Client"/>
    <bpmn:participant name="Search Entity SP" processRef="search_entity_sp"/>
    <bpmn:messageFlow id="search_entity_request_message_flow" name="Search Entity
Request MessageFlow" sourceRef="search_entity_client"
targetRef="search_entity_start"/>

```

```

    <bpmn:messageFlow id="search_entity_reject_message_flow" name="Search Entity
Reject MessageFlow" sourceRef="search_entity_reject_request"
targetRef="search_entity_client"/>
    <bpmn:messageFlow id="search_entity_response_message_flow" name="Search
Entity Response MessageFlow" sourceRef="search_entity_fulfill_request"
targetRef="search_entity_client"/>
  </bpmn:collaboration>
  <bpmn:process id="search_entity_sp" processType="Public"
definitionalCollaborationRef="search_entity_collaboration">
    <bpmn:startEvent id="search_entity_start">
      <bpmn:messageEventDefinition/>
    </bpmn:startEvent>
    <bpmn:sequenceFlow sourceRef="search_entity_start"
targetRef="search_entity_fulfill_request"/>
    <bpmn:task id="search_entity_fulfill_request" name="Fulfill Request"/>
    <bpmn:boundaryEvent id="search_entity_fulfill_error_occurrence"
attachedToRef="search_entity_fulfill_request">
      <bpmn:eventDefinition xsi:type="bpmn:tErrorEventDefinition"/>
    </bpmn:boundaryEvent>

    <bpmn:sequenceFlow sourceRef="search_entity_fulfill_error_occurrence"
targetRef="search_entity_reject_request"/>
    <bpmn:sendTask id="search_entity_reject_request" name="Reject Request"/>
    <bpmn:sequenceFlow sourceRef="search_entity_reject_request"
targetRef="search_entity_error_end"/>
    <bpmn:endEvent id="search_entity_error_end" name="Error">
      <bpmn:eventDefinition xsi:type="bpmn:tErrorEventDefinition"/>
    </bpmn:endEvent>

    <bpmn:sequenceFlow sourceRef="search_entity_fulfill_request"
targetRef="search_entity_respond"/>
    <bpmn:sendTask id="search_entity_respond" name="Respond"/>
    <bpmn:sequenceFlow sourceRef="search_entity_respond"
targetRef="search_entity_success_end"/>
    <bpmn:endEvent id="search_entity_success_end" name="Success"/>
  </bpmn:process>

  <!-- ***** -->
  <!--           Messages and ItemDefinitions for Search Content           -->
  <!-- ***** -->
  <bpmn:message id="search_content_request" name="SearchContentRequest"
itemRef="search_content_request_itemdef">
    <bpmn:extensionElements>

<bpmnext1:MessageFlowRef>search_content_request_message_flow</bpmnext1:MessageFlo
wRef>
    </bpmn:extensionElements>
  </bpmn:message>
  <bpmn:message id="search_content_response" name="SearchContentResponse"
itemRef="search_content_response_itemdef">
    <bpmn:extensionElements>

<bpmnext1:MessageFlowRef>search_content_reject_message_flow</bpmnext1:MessageFlow
Ref>

<bpmnext1:MessageFlowRef>search_content_response_message_flow</bpmnext1:MessageFl
owRef>
    </bpmn:extensionElements>
  </bpmn:message>

```

```

    <bpmn:itemDefinition id="search_content_request_itemdef"
    structureRef="mpegm:SearchContentRequest"/><!-- Binding of the bpmn:message to
    the actual MPEG-M request message definition. -->
    <bpmn:itemDefinition id="search_content_response_itemdef"
    structureRef="mpegm:SearchContentResponse"/><!-- Binding of the bpmn:message to
    the actual MPEG-M response message definition. -->
  </bpmn:definitions>

```

5.2 Base data types, elements, and common messages

5.2.1 Introduction

This subclause specifies common base data types and elements that are used throughout various MPEG-M Services.

The common base data types and elements are defined in a separate XML Schema definition with the target namespace `urn:mpeg:mpegM:schema:01-base-NS:2012`, facilitating their reuse outside of ISO/IEC 23006.

NOTE The default namespace used in this subclause is `urn:mpeg:mpegM:schema:01-base-NS:2012`.

5.2.2 MPEG-M base 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 are gathered in a same document with the following declaration defining in particular the target namespace and the namespaces prefixes.

```

<?xml version="1.0" encoding="UTF-8"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:mpegmb="urn:mpeg:mpegM:schema:01-base-NS:2012"
  xmlns:didl="urn:mpeg:mpeg21:2006:07-DIDL-NS"
  xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
  xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
  xmlns:cel="urn:mpeg:mpeg21:cel:core:2012"
  xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
  xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
  xmlns:ipmpinfo-msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  targetNamespace="urn:mpeg:mpegM:schema:01-base-NS:2012"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="ISO/IEC 23006-4 2nd edition" id="mpeg-m-base.xsd">
  <import namespace="http://www.w3.org/2000/09/xmldsig#"
  schemaLocation="w3c/xmldsig-core-schema.xsd"/>
  <import namespace="urn:mpeg:mpeg21:2006:07-DIDL-NS"
  schemaLocation="mpeg/didl.xsd"/>
  <import namespace="urn:mpeg:mpeg21:2002:01-DII-NS"
  schemaLocation="http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-
  21_schema_files/dii/dii.xsd"/>
  <import namespace="urn:mpeg:mpeg7:schema:2004" schemaLocation="mpeg/mpeg7-
  v3.xsd"/>
  <import namespace="urn:mpeg:mpeg21:cel:core:2012" schemaLocation="cel-
  core.xsd"/>

```

```

<import namespace="urn:mpeg:mpeg21:2003:01-REL-R-NS" schemaLocation="mpeg/rel-
r.xsd"/>
<import namespace="urn:mpeg:mpeg21:2003:01-REL-SX-NS" schemaLocation="mpeg/rel-
sx.xsd"/>
<import namespace="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
schemaLocation="mpeg/ipmpinfo-msx.xsd"/>
<import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
schemaLocation="oasis/saml-schema-assertion-2.0.xsd"/>

```

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

```
</schema>
```

5.2.3 Common elements and types

5.2.3.1 Introduction

This Subclause describes elements and complex element types which are used throughout various Services.

5.2.3.2 ServiceTypeType

5.2.3.2.1 Introduction

The *ServiceTypeType* defines an XML representation for the concept of Service Types described in 5.1.4.

5.2.3.2.2 Syntax

```

<!-- ##### -->
<!-- Service Type -->
<!-- ##### -->
<complexType name="ServiceTypeType">
  <sequence>
    <element ref="mpegmb:ServiceTypeIdentifier"/>
    <element name="Name" type="string"/>
    <element name="RequestParametersType" type="QName" minOccurs="0"/>
    <element name="CompletionParametersType" type="QName" minOccurs="0"/>
  </sequence>
</complexType>

```

5.2.3.2.3 Semantics

Semantics of the *ServiceTypeType*:

<i>Name</i>	<i>Definition</i>
<i>ServiceTypeType</i>	Top-level type for the specification of a Service Type.
<i>ServiceTypeIdentifier</i>	Unique identifier of the Service Type.

Name	Descriptive name of the Service Type.
RequestParametersType	Specifies the complex type for the RequestParameters element of the ProcessContentRequest message.
CompletionParametersType	Specifies the complex type for the CompletionParameters element of the ProcessContentCompletion message.

5.2.3.2.4 Additional validation rules

5.2.3.2.4.1 Introduction

For the purpose of referencing the additional validation rules are numbered.

5.2.3.2.4.2 The value of the RequestParametersType element must be an existing complex type that extends RequestParametersBaseType.

5.2.3.2.4.3 The value of the CompletionParametersType element must be an existing complex type that extends CompletionParametersBaseType.

5.2.3.3 ContractIdentifier

5.2.3.3.1 Introduction

Contracts within MPEG-M are identified by the ContractIdentifier element.

NOTE This Part of ISO/IEC 23006 does neither specify a particular identification scheme for Contracts nor a mechanism for attaching the ContractIdentifier element to a Contract.

5.2.3.3.2 Syntax

```
<!-- Definition of ContractIdentifier -->
<element name="ContractIdentifier" type="mpeg7:UniqueIDType" />
```

5.2.3.3.3 Semantics

Semantics of the ContractIdentifier:

Name	Definition
ContractIdentifier	Describes the identification of a Contract based on the mpeg7:UniqueIDType. The contract identifier corresponds to the cel:contractId.

5.2.3.4 LicenseIdentifier

5.2.3.4.1 Introduction

Licenses within MPEG-M are identified by the LicenseIdentifier element.

NOTE This Part of ISO/IEC 23006 does neither specify a particular identification scheme for Licenses nor a mechanism for attaching the LicenseIdentifier element to a License.

5.2.3.4.2 Syntax

```
<!-- Definition of LicenseIdentifier -->
<element name="LicenseIdentifier" type="mpeg7:UniqueIDType" />
```

5.2.3.4.3 Semantics

Semantics of the `LicenseIdentifier`:

<i>Name</i>	<i>Definition</i>
<code>LicenseIdentifier</code>	Describes the identification of a License. The license identifier corresponds to the <code>rel-r:licenseId</code> .

5.2.3.5 ServiceTypeIdentifier

5.2.3.5.1 Introduction

Service Types within MPEG-M are identified by the `ServiceTypeIdentifier` element.

NOTE The namespace URIs for XML Schemas of Service Types specified in 4.1.4 also serve as Service Type Identifiers of the Service Types defined in this Part of ISO/IEC 23006.

5.2.3.5.2 Syntax

```
<!-- Definition of ServiceTypeIdentifier -->
<element name="ServiceTypeIdentifier" type="mpeg7:UniqueIDType" />
```

5.2.3.5.3 Semantics

Semantics of the `ServiceTypeIdentifier`:

<i>Name</i>	<i>Definition</i>
<code>ServiceTypeIdentifier</code>	Describes the identification of a Service Type.

5.2.3.6 SIDIdentifier

5.2.3.6.1 Introduction

Service Instance Declarations (SIDs) within MPEG-M are identified by the `SIDIdentifier` element.

NOTE This Part of ISO/IEC 23006 does not specify a particular identification scheme for SIDs.

5.2.3.6.2 Syntax

```
<!-- Definition of SIDIdentifier -->
<element name="SIDIdentifier" type="mpeg7:UniqueIDType" />
```

5.2.3.6.3 Semantics

Semantics of the `SIDIdentifier`:

<i>Name</i>	<i>Definition</i>
<code>SIDIdentifier</code>	Describes the identification of a Service Instance Declaration.

5.2.3.7 UserIdentifier

5.2.3.7.1 Introduction

Users within MPEG-M are identified by the `UserIdentifier` element.

NOTE This Part of ISO/IEC 23006 does not specify a particular identification scheme for Users.

5.2.3.7.2 Syntax

```
<!-- Definition of UserIdentifier -->
<element name="UserIdentifier" type="mpeg7:UniqueIDType" />
```

5.2.3.7.3 Semantics

Semantics of the `UserIdentifier`:

<i>Name</i>	<i>Definition</i>
<code>UserIdentifier</code>	Describes the identification of a User.

5.2.3.8 EntityBaseType

5.2.3.8.1 Introduction

Throughout MPEG-M, Entities are in general included in messages by one of the following mechanisms:

- **Direct inclusion:** The Entity is typically included as an XML infoset (if applicable).
- **By reference:** One or more locations, represented as URLs, are specified, where the Entity can be found. If multiple URLs are given, they must all link to replications of the same Entity.
- **By identifier:** One or more identifiers specifying the same Entity. The resolution of identifiers is out of the scope of MPEG-M.

NOTE This Part of ISO/IEC 23006 does not mandate the use of particular inclusion mechanisms for Elementary Services. If a particular inclusion mechanism is recommended for an Elementary Service, this is stated in the General Service Definition. For example, for an `IdentifyContentRequest` message, direct inclusion or inclusion by reference should be preferred over inclusion by identifier. On the other hand, for a `RevokeContractRequest` message, inclusion by identifier is preferred over inclusion by reference because all replications of that contract shall be revoked and not just the one referenced instance.

The representations of Entity inclusions are derived from the `EntityBaseType`.

5.2.3.8.2 Syntax

```

<!-- Definition of EntityBaseType -->
<complexType name="EntityBaseType" abstract="true">
  <annotation>
    <documentation xml:lang="en">
      Derived types append a choice between direct inclusion, inclusion by
      reference, and inclusion by identifier.
    </documentation>
  </annotation>
</complexType>

```

5.2.3.8.3 Semantics

Semantics of the EntityBaseType:

<i>Name</i>	<i>Definition</i>
EntityBaseType	Base type for containers that include a specific kind of Entity by either direct inclusion, inclusion by reference, or inclusion by identifier.

5.2.3.9 ContentEntityType

5.2.3.9.1 Introduction

Throughout MPEG-M, Content is in general either referenced or included in the ContentEntityType. Content is represented as MPEG-21 DIDL. Annex F specifies a Content representation (called MPEG-M DIDL) based on MPEG-21 DIDL. The use of MPEG-M DIDL is optional.

5.2.3.9.2 Syntax

```

<complexType name="ContentEntityType">
  <complexContent>
    <extension base="mpegmb:EntityBaseType">
      <choice>
        <element ref="didl:DIDL"/>
        <element name="ContentRef" type="anyURI" maxOccurs="unbounded"/>
        <element ref="dii:Identifier" maxOccurs="unbounded"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

5.2.3.9.3 Semantics

Semantics of the ContentEntityType:

<i>Name</i>	<i>Definition</i>
ContentEntityType	Container for a Digital Item, a reference to it, or an identifier of it.

<code>didl:DIDL</code>	Digital Item Declaration of the Content. The Content may also contain sensory information as specified in ISO/IEC 23005-3. NOTE Sensory Information is considered as content (not as metadata) and is therefore represented as a <code>didl:Resource</code> inside a <code>didl:Component</code> of a Digital Item. NOTE Annex F specifies a Content representation (called MPEG-M DIDL) based on MPEG-21 DIDL. The use of MPEG-M DIDL is optional.
<code>ContentRef</code>	Reference to the Content. <code>ContentRef</code> should specify a location rather than an Identifier. Multiple locations for the same Content can be specified.
<code>dii:Identifier</code>	Identifier of the Content as specified in ISO/IEC 21000-3. Multiple identifiers for the same Content can be specified.

5.2.3.9.4 Additional validation rules

5.2.3.9.4.1 Introduction

For the purpose of referencing the additional validation rules are numbered.

5.2.3.9.4.2 If multiple `ContentRef` elements are specified, they must all point to copies of the same Content.

5.2.3.9.4.3 If multiple `dii:Identifier` elements are specified, they must all identify the same Content.

5.2.3.10 ContractEntityType

5.2.3.10.1 Introduction

Throughout MPEG-M, Contracts are in general either referenced or included in the `ContractEntityType`. Contracts are represented as ISO/IEC 21000-20 `cel:contract`.

5.2.3.10.2 Syntax

```
<complexType name="ContractEntityType">
  <complexContent>
    <extension base="mpegm:EntityType">
      <choice>
        <element ref="cel:contract"/>
        <element name="ContractRef" type="anyURI" maxOccurs="unbounded"/>
        <element ref="mpegmb:ContractIdentifier" maxOccurs="unbounded"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

5.2.3.10.3 Semantics

Semantics of the `ContractEntityType`:

<i>Name</i>	<i>Definition</i>
<code>ContractEntityType</code>	Container for a Contract or a reference to it.
<code>cel:contract</code>	MPEG-21 CEL contract.
<code>ContractRef</code>	Reference to the Contract. <code>ContractRef</code> specifies a location rather than an Identifier, matching the <code>cel:contractId</code> element. Several locations for the same Contract can be specified.
<code>ContractIdentifier</code>	Identifier of the Contract. Multiple identifiers for the same Contract can be specified.

5.2.3.10.4 Additional validation rules

5.2.3.10.4.1 Introduction

For the purpose of referencing the additional validation rules are numbered.

5.2.3.10.4.2 If multiple `ContractRef` elements are specified, they must all point to copies of the same Contract.

5.2.3.10.4.3 If multiple `ContractIdentifier` elements are specified, they must all identify the same Contract.

5.2.3.11 DeviceEntityType

5.2.3.11.1 Introduction

Throughout MPEG-M, Devices are in general either referenced or included in the `DeviceEntityType`. Devices are represented as ISO/IEC 23000-5 `ipmpinfo-msx:ToolBody`.

5.2.3.11.2 Syntax

```
<complexType name="DeviceEntityType">
  <complexContent>
    <extension base="mpegm:EntityType">
      <choice>
        <element ref="ipmpinfo-msx:ToolBody"/>
        <element name="DeviceRef" type="anyURI" maxOccurs="unbounded"/>
        <element ref="ipmpinfo-msx:ToolBodyID" maxOccurs="unbounded"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

5.2.3.11.3 Semantics

Semantics of the `DeviceEntityType`:

<i>Name</i>	<i>Definition</i>
<code>DeviceEntityType</code>	Container for a Device or a reference to it.
<code>ipmpinfo-msx:ToolBody</code>	A software Device.
<code>DeviceRef</code>	Reference to the Device. <code>DeviceRef</code> should specify a location rather than an Identifier. Several locations for the same Device can be specified.
<code>ipmpinfo-msx:ToolBodyID</code>	Identifier of the Device. Multiple identifiers for the same Device can be specified.

5.2.3.11.4 Additional validation rules

5.2.3.11.4.1 Introduction

For the purpose of referencing the additional validation rules are numbered.

5.2.3.11.4.2 If multiple `DeviceRef` elements are specified, they must all point to the same Device.

5.2.3.11.4.3 If multiple `ipmpinfo-msx:ToolBodyID` elements are specified, they must all identify the same Device.

5.2.3.12 LicenseEntityType

5.2.3.12.1 Introduction

Throughout MPEG-M, Licenses are in general either referenced or included in the `LicenseEntityType`. Licenses are represented as ISO/IEC 21000-5 `rel-r:license`.

5.2.3.12.2 Syntax

```
<complexType name="LicenseEntityType">
  <complexContent>
    <extension base="mpegm:EntityType">
      <choice>
        <element ref="rel-r:license"/>
        <element name="LicenseRef" type="anyURI" maxOccurs="unbounded"/>
        <element ref="mpegmb:LicenseIdentifier" maxOccurs="unbounded"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

5.2.3.12.3 Semantics

Semantics of the `LicenseEntityType`:

<i>Name</i>	<i>Definition</i>
<code>LicenseEntityType</code>	Container for a License or a reference to it.
<code>rel-r:license</code>	MPEG-21 REL license.
<code>LicenseRef</code>	Reference to the License. <code>LicenseRef</code> should specify a location rather than an Identifier. Several locations for the same License can be specified.
<code>LicenseIdentifier</code>	Identifier of the License. Multiple identifiers for the same License can be specified.

5.2.3.12.4 Additional validation rules

5.2.3.12.4.1 Introduction

For the purpose of referencing the additional validation rules are numbered.

5.2.3.12.4.2 If multiple `LicenseRef` elements are specified, they must all point to copies of the same License.

5.2.3.12.4.3 If multiple `LicenseIdentifier` elements are specified, they must all identify the same License.

5.2.3.13 ServiceTypeEntityType

5.2.3.13.1 Introduction

Throughout MPEG-M, Service Types are in general either referenced or included in the `ServiceTypeEntityType`. Service Types are represented as `ServiceTypeType`, defined in 5.2.3.2.

5.2.3.13.2 Syntax

```
<complexType name="ServiceTypeEntityType">
  <complexContent>
    <extension base="mpegm:EntityType">
      <choice>
        <element name="ServiceType" type="mpegm:ServiceTypeType"/>
        <element name="ServiceTypeRef" type="anyURI" maxOccurs="unbounded"/>
        <element ref="mpegmb:ServiceTypeIdentifier" maxOccurs="unbounded"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

5.2.3.13.3 Semantics

Semantics of the `ServiceEntityType`:

<i>Name</i>	<i>Definition</i>
<code>ServiceEntityType</code>	Container for a Service Type or a reference to it.
<code>ServiceType</code>	Service Type provided as inline definition.
<code>ServiceTypeRef</code>	Reference to the Service Type. <code>ServiceTypeRef</code> should specify a location rather than an Identifier. Several locations for the same Service Type can be specified.
<code>ServiceTypeIdentifier</code>	Identifier of the Service Type. Multiple identifiers for the same Service Type can be specified.

5.2.3.13.4 Additional validation rules

5.2.3.13.4.1 Introduction

For the purpose of referencing the additional validation rules are numbered.

5.2.3.13.4.2 If multiple `ServiceTypeRef` elements are specified, they must all point to the same Service Type.

5.2.3.13.4.3 If multiple `ServiceTypeIdentifier` elements are specified, they must all identify the same Service Type.

5.2.3.14 UserEntityType

5.2.3.14.1 Introduction

Throughout MPEG-M, Users are in general either referenced or included in the `UserEntityType`. Users are represented as ISO/IEC 15938-5 `mpeg7:AgentType`.

5.2.3.14.2 Syntax

```
<complexType name="UserEntityType">
  <complexContent>
    <extension base="mpegm:EntityBaseType">
      <choice>
        <element name="User" type="mpeg7:AgentType"/>
        <element name="UserRef" type="anyURI" maxOccurs="unbounded"/>
        <element ref="mpegmb:UserIdentifier" maxOccurs="unbounded"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

5.2.3.14.3 Semantics

Semantics of the `UserEntityType`:

<i>Name</i>	<i>Definition</i>
<code>UserEntityType</code>	Container for a User or a reference to it.
<code>User</code>	Description of a User provided as inline definition.
<code>UserRef</code>	Reference to the User. <code>UserRef</code> should specify a location rather than an Identifier. Several locations for the same User can be specified.
<code>UserIdentifier</code>	Identifier of the User. Multiple identifiers for the same User can be specified.

5.2.3.14.4 Additional validation rules**5.2.3.14.4.1 Introduction**

For the purpose of referencing the additional validation rules are numbered.

5.2.3.14.4.2 If multiple `UserRef` elements are specified, they must all point to the same User.

5.2.3.14.4.3 If multiple `UserIdentifier` elements are specified, they must all identify the same User.

5.2.3.15 RequestParametersBaseType**5.2.3.15.1 Introduction**

The request parameters of all Service Types of the Process Content Elementary Service (see 5.16.1) are based on the `RequestParametersBaseType`.

5.2.3.15.2 Syntax

```
<!-- Definition of RequestParametersBaseType -->
<complexType name="RequestParametersBaseType" abstract="true"/>
```

5.2.3.15.3 Semantics

Semantics of the `RequestParametersBaseType`:

<i>Name</i>	<i>Definition</i>
<code>RequestParametersBaseType</code>	Base type for all request parameters of the Process Content Elementary Service.

5.2.3.16 CompletionParametersBaseType

5.2.3.16.1 Introduction

The completion parameters of all Service Types of the Process Content Elementary Service (see 5.16.1) are based on the CompletionParametersBaseType.

5.2.3.16.2 Syntax

```
<!-- Definition of CompletionParametersBaseType -->
<complexType name="CompletionParametersBaseType" abstract="true"/>
```

5.2.3.16.3 Semantics

Semantics of the CompletionParametersBaseType:

<i>Name</i>	<i>Definition</i>
CompletionParametersBaseType	Base type for all completion parameters of the Process Content Elementary Service.

5.2.3.17 SupportedMetadataSchemaType

5.2.3.17.1 Introduction

Depending on the strictness of metadata support indicated in the SID (see Annex C), various metadata representations can be deployed in MPEG-M. Users can indicate supported metadata schemas and their respective preferences through the SupportedMetadataSchemaType.

5.2.3.17.2 Syntax

```
<!-- Definition of SupportedMetadataSchemaType -->
<complexType name="SupportedMetadataSchemaType">
  <simpleContent>
    <extension base="anyURI">
      <attribute name="pref" type="mpeg7:zeroToOneType" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

5.2.3.17.3 Semantics

Semantics of the SupportedMetadataSchemaType:

<i>Name</i>	<i>Definition</i>
SupportedMetadataSchemaType	The <code>xsd:targetNamespace</code> of a supported metadata schema for the Entity description.
pref	Indicates the User's preference of this metadata schema. Higher numbers indicate higher preferences.

5.2.3.18 SupportedTransferProtocolType

5.2.3.18.1 Introduction

Users can indicate supported transfer protocols and their respective preferences through the SupportedTransferProtocolType.

5.2.3.18.2 Syntax

```
<!-- Definition of SupportedTransferProtocolType -->
<complexType name="SupportedTransferProtocolType">
  <sequence>
    <element name="TransferProtocol" type="mpeg7:ControlledTermUseType">
      <annotation>
        <documentation xml:lang="en">
          Proposed Classification Scheme: urn:mpeg:mpegM:cs:03-es-
NS:2012:TransferProtocolCS
        </documentation>
      </annotation>
    </element>
    <element name="Option" type="mpegmb:KeyValueDataType" minOccurs="0"
maxOccurs="unbounded"/>
  </sequence>
  <attribute name="pref" type="mpeg7:zeroToOneType"/>
</complexType>
```

5.2.3.18.3 Semantics

Semantics of the SupportedTransferProtocolType:

<i>Name</i>	<i>Definition</i>
SupportedTransferProtocolType	Indication of a supported transfer protocol and its preference.
TransferProtocol	Specifies the transfer protocol using an MPEG-7 Classification Scheme. Terms for the TransferProtocol are specified by the TransferProtocolCS (urn:mpeg:mpegM:cs:03-es-NS:2012:TransferProtocolCS) in G.4.
Option	Application-specific options for this transfer protocol.
pref	Indicates the User's preference of this transfer protocol. Higher numbers indicate higher preferences.

5.2.3.19 RevocationReasonType

5.2.3.19.1 Introduction

The RevocationReasonType allows for the specification of a reason for a revocation.

5.2.3.19.2 Syntax

```

<!-- Definition of RevocationReasonType -->
<simpleType name="RevocationReasonType">
  <union>
    <simpleType>
      <restriction base="NMTOKEN">
        <enumeration value="TERMINATE"/>
        <enumeration value="CANCEL"/>
        <enumeration value="ABORT"/>
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="mpeg7:termReferenceType"/>
    </simpleType>
  </union>
</simpleType>

```

5.2.3.19.3 Semantics

Semantics of the `RevocationReasonType`:

Name	Definition
<code>RevocationReasonType</code>	Reason of a revocation chosen from a predefined set: TERMINATE, CANCEL, or ABORT. The possible values are defined in Table 6. Other values that are datatype-valid with respect to <code>mpeg7:termReferenceType</code> are reserved.
<code>OtherReason</code>	Descriptive information of any other revocation reason.

The reason of revocation can be one of the following:

Table 6 — Possible values of the `RevocationType`

Protocol Code	Definition
TERMINATE	Terminate naturally the Entity (i.e., contract or license). The requester must be entitled to do it, otherwise it would be making an abortion.
CANCEL	Ask for an agreed cancellation of an Entity (i.e., contract or license).
ABORT	Abort the Entity (i.e., contract or license) unilaterally, being then subject to the penalties for abortion foreseen in the breachment clauses.

5.2.3.20 EntityOrderInformationType

5.2.3.20.1 Introduction

The `EntityOrderInformationType` comprises information about an order in a transaction.

5.2.3.20.2 Syntax

```

<!-- Definition of EntityOrderInformationType -->
<complexType name="EntityOrderInformationType" abstract="true">
  <sequence>
    <annotation>
      <documentation xml:lang="en">
        Each extension puts here a transacted Entity of the appropriate type
        (e.g., mpegmb:ContentEntityType).
      </documentation>
    </annotation>
    <element name="OrderID" type="anyURI"/>
    </element>
    <element name="Buyer" type="mpegmb:UserEntityType"/>
    <element name="TransactionDate" type="dateTime" minOccurs="0"/>
    <sequence minOccurs="0">
      <element name="Amount" type="rel-sx:Rate"/>
      <element name="VAT" type="rel-sx:Rate" minOccurs="0"/>
    </sequence>
  </sequence>
</complexType>

```

5.2.3.20.3 Semantics

Semantics of the EntityOrderInformationType:

<i>Name</i>	<i>Definition</i>
OrderInformationType	Collection of information about the order in a transaction.
OrderID	Identifier of the order.
Buyer	Information on the buyer.
TransactionDate	Date when the transaction was completed. Only used if the transaction has already been completed.
Amount	Amount of the transaction (including any VAT).
VAT	Value Added Tax (VAT) of the transaction.

5.2.3.21 ContentOrderInformationType

5.2.3.21.1 Introduction

The ContentOrderInformationType comprises information about a content order in a transaction.

5.2.3.21.2 Syntax

```

<!-- Definition of ContentOrderInformationType -->
<complexType name="ContentOrderInformationType">
  <complexContent>
    <extension base="mpegmb:EntityOrderInformationType">
      <sequence>
        <element name="TransactedContent" type="mpegmb:ContentEntityType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

5.2.3.21.3 Semantics

Semantics of the ContentOrderInformationType:

<i>Name</i>	<i>Definition</i>
ContentOrderInformationType	Collection of information about the content order in a transaction.
TransactedContent	Content being transacted.

5.2.3.22 LicenseOrderInformationType

5.2.3.22.1 Introduction

The LicenseOrderInformationType comprises information about a license order in a transaction.

5.2.3.22.2 Syntax

```

<!-- Definition of LicenseOrderInformationType -->
<complexType name="LicenseOrderInformationType">
  <complexContent>
    <extension base="mpegmb:EntityOrderInformationType">
      <sequence>
        <element name="TransactedLicense" type="mpegmb:LicenseEntityType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

5.2.3.22.3 Semantics

Semantics of the LicenseOrderInformationType:

<i>Name</i>	<i>Definition</i>
LicenseOrderInformationType	Collection of information about the license order in a transaction.
TransactedLicense	License being transacted.

5.2.4 Base Protocol

5.2.4.1 Introduction

All the complex types of messages exchanged in protocols for MPEG-M Services extend from the complex type `ProtocolBaseType`. The `ProtocolBaseType` defines elements that are used in all protocol messages.

5.2.4.2 Syntax

```
<!-- Definition of ProtocolBaseType. Base type for all protocol messages. -->
<complexType name="ProtocolBaseType" abstract="true">
  <sequence>
    <element name="TransactionIdentifier" type="string"/>
    <element name="Timestamp" type="dateTime" minOccurs="0"/>
    <element name="Entry" type="mpegmb:KeyValueDataType" minOccurs="0"
maxOccurs="unbounded"/>
    <element name="ApplicationSpecificData"
type="mpegmb:ApplicationSpecificDataType" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="dsig:Signature" minOccurs="0"/>
  </sequence>
</complexType>

<complexType name="KeyValueDataType">
  <simpleContent>
    <extension base="string">
      <attribute name="key" type="mpeg7:termReferenceType" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="ApplicationSpecificDataType">
  <sequence>
    <any namespace="##other" processContents="lax"/><!-- May contain any
application-specific XML element. -->
  </sequence>
</complexType>
```

5.2.4.3 Semantics

Semantics of the `ProtocolType`:

<i>Name</i>	<i>Definition</i>
<code>ProtocolType</code>	Base type for all the protocol message types.
<code>TransactionIdentifier</code>	Conveys a value which is used to track the relation between exchanged messages. Any message in response to another message shall specify the same <code>TransactionIdentifier</code> value contained in the request.
<code>Timestamp</code>	The precise date and time when the message is sent.
<code>Entry</code>	An arbitrary property-value pair for application-specific data.
<code>ApplicationSpecificData</code>	Other application-specific data.

`dsig:Signature` Signature of the protocol message. If the signature is present, it signs the entire protocol message and not just the elements of the `ProtocolType`.

Semantics of the `KeyValue` type:

Name	Definition
<code>KeyValue</code>	Type for an application-specific property-value pair. The value is contained in the text node of the element.
<code>key</code>	Identifying the property of this property-value pair. The key references an MPEG-7 term using a URI. The term should be defined in an MPEG-7 Classification Scheme.
NOTE Implementations may define their own application-specific Classification Schemes.	

Semantics of the `ApplicationSpecificData` type:

Name	Definition
<code>ApplicationSpecificData</code>	Container for any application-specific XML-based data structures.

5.2.4.4 Examples

EXAMPLE 1 The example below shows an `Entry` element that specifies the start time for a Service session. A client may specify such a start time in the request message in order to instruct the SP to start the processing of the request at a certain time.

```
<Entry key="urn:example:time-cs:start-time">2011-03-16T14:28:00</Entry>
```

EXAMPLE 2 The example below shows an `ApplicationSpecificData` element that specifies a Service Level Agreement (SLA) for the Service session. The client may specify additional requirements such as network properties, transport mechanism, timeout for delivery, size of content to be delivered, etc. for Deliver Content.

```
<ApplicationSpecificData>
  <ns1:SLA>
    <ns1:Timeout>PT3M30S</ns1:Timeout>
    <!-- Specification of further requirements... -->
  </ns1:SLA>
</ApplicationSpecificData>
```

5.2.5 Common messages

5.2.5.1 Introduction

Many of the protocols define a request and a response messages. Some of the elements in this message are common and they can be abstracted.

Response message make use of either a *success acknowledgement* (represented by the `ProtocolSuccessType`) or an *error information* (represented by the `ProtocolFailureType`).

All protocols, except the Authenticate User elementary service defined in 5.5.3.5, require that Client and Service Provider mutually authenticate before the first protocol step. This authentication may be either explicit (e.g., authentication protocol) or implicit (e.g., trusted environment). This Part of ISO/IEC 23006 does not specify the mechanism of the authentication. One possible authentication mechanism is the Authenticate User elementary service as defined in 5.5.3.5. Note that ISO/IEC 23006-2 specifies profiles for the authentication mechanism.

5.2.5.2 Syntax

```

<!-- Definition of ProtocolRequestType -->
<complexType name="ProtocolRequestType" abstract="true">
  <complexContent>
    <extension base="mpegmb:ProtocolBaseType">
      <sequence>
        <element ref="saml:Assertion" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Definition of ProtocolResponseType -->
<complexType name="ProtocolResponseType" abstract="true">
  <complexContent>
    <extension base="mpegmb:ProtocolBaseType">
      <!-- Each Elementary Service appends here the choice between
[ServiceName]Success and [ServiceName]Failure. -->
    </extension>
  </complexContent>
</complexType>

<complexType name="ProtocolSuccessType">
  <sequence>
    <element name="SuccessCode" type="mpegmb:SuccessCodeType" minOccurs="0"/>
    <element name="DisplayString" type="string" minOccurs="0"/>
  </sequence>
</complexType>

<simpleType name="SuccessCodeType">
  <union>
    <simpleType>
      <restriction base="NMTOKEN">
        <enumeration value="OK"/>
        <enumeration value="CREATED"/>
        <enumeration value="ACCEPTED"/>
        <enumeration value="NON-AUTHORITATIVE_CONTENT"/>
        <enumeration value="NO_CONTENT"/>
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="mpeg7:termReferenceType"/>
    </simpleType>
  </union>
</simpleType>

<complexType name="ProtocolFailureType">
  <sequence>
    <element name="FailureCode" type="mpegmb:FailureCodeType" minOccurs="0"/>
    <element name="DisplayString" type="string" minOccurs="0"/>
  </sequence>

```