INTERNATIONAL STANDARD

ISO/IEC 23005-3

Second edition
2013-04-15

Information technology — Media context and control —

Part 3: Sensory information

Technologies de l'information — Contrôle et contexte de supports —

Partie 3: Information sensorielle

Reference number
ISO/IEC 23005-3:2013(E)

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

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 23005-3 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 23005-3:2011), which has been technically revised.

ISO/IEC 23005 consists of the following parts, under the general title *Information technology — Media context and control*:

— *Part 1: Architecture*

— *Part 2: Control information*

— *Part 3: Sensory information*

— *Part 4: Virtual world object characteristics*

— *Part 5: Data formats for interaction devices*

— *Part 6: Common types and tools*

— *Part 7: Conformance and reference software*

Introduction

ISO/IEC 23005 (MPEG-V) provides an architecture and specifies associated information representations to enable interoperability between virtual worlds, e.g. digital content provider of a virtual world, (serious) gaming, simulation, DVD, and with the real world, e.g. sensors, actuators, vision and rendering, robotics (e.g. for revalidation), (support for) independent living, social and welfare systems, banking, insurance, travel, real estate, rights management and many others.

Virtual worlds (often referred to as 3D3C for 3D visualization & navigation and the 3C's of Community, Creation and Commerce) integrate existing and emerging (media) technologies (e.g. instant messaging, video, 3D, VR, AI, chat, voice, etc.) that allow for the support of existing, and the development of new kinds of, social networks. The emergence of virtual worlds as platforms for social networking is recognized by businesses as an important issue for at least two reasons:

— It offers the power to reshape the way companies interact with their environments (markets, customers, suppliers, creators, stakeholders, etc.) in a fashion comparable to the Internet.

— It allows for the development of new (breakthrough) business models, services, applications and devices.

Each virtual world however has a different culture and audience making use of these specific worlds for a variety of reasons. These differences in existing metaverses permit users to have unique experiences. Resistance to real-world commercial encroachment still exists in many virtual worlds, where users primarily seek an escape from real life. Hence, marketers should get to know a virtual world beforehand and the rules that govern each individual universe.

Although realistic experiences have been achieved via devices such as 3-D audio/visual devices, it is hard to realize sensory effects only with the presentation of audiovisual contents. The addition of sensory effects leads to even more realistic experiences in the consumption of audiovisual contents. This will lead to the application of new media for enhanced experiences of users in a more realistic sense.

Such new media will benefit from the standardization of control and sensory information which can include sensory effect metadata, sensory device capabilities/commands, user sensory preferences, and various delivery formats. The MPEG-V architecture can be applicable for various business models for which audiovisual contents can be associated with sensory effects that need to be rendered on appropriate sensory devices.

This part of ISO/IEC 23005 contains the sensory information which can stimulate other senses than vision or audition, e.g. olfaction, mechanoreception, equilibrioception, or thermoception. That is, in addition to the audio-visual content of, e.g., a movie, also other sense shall be stimulated giving her/him the sensation of being part of the particular media which shall result in a worthwhile, informative user experience.

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

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

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

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

Information technology — Media context and control —

Part 3:
Sensory information

1 Scope

This part of ISO/IEC 23005 specifies syntax and semantics of description schemes and descriptors that represent sensory information. This part of ISO/IEC 23005 enhances the experience of users while consuming media resources by stimulating human multi-sensor such as tactile, orfactory, light sense, temperature sense, etc.

The system architecture is depicted in Figure 1 and the scope of this part of ISO/IEC 23005 is highlighted. That is, only the information representation that acts as an input to the possible Adaptation VR – as defined in ISO/IEC 23005-1 – is specified in this part of ISO/IEC 23005.

The adaptation engine for Sensory Information is to adapt Sensory information to Device Command which is the actual signal to control multi-sensory devices as defined in ISO/IEC 23005-5. This adaptation process is not mandatory in case the sensory information may directly control the actual devices.

NOTE 1 The actual Adaptation VR is deliberately informative and left open for industry competition.

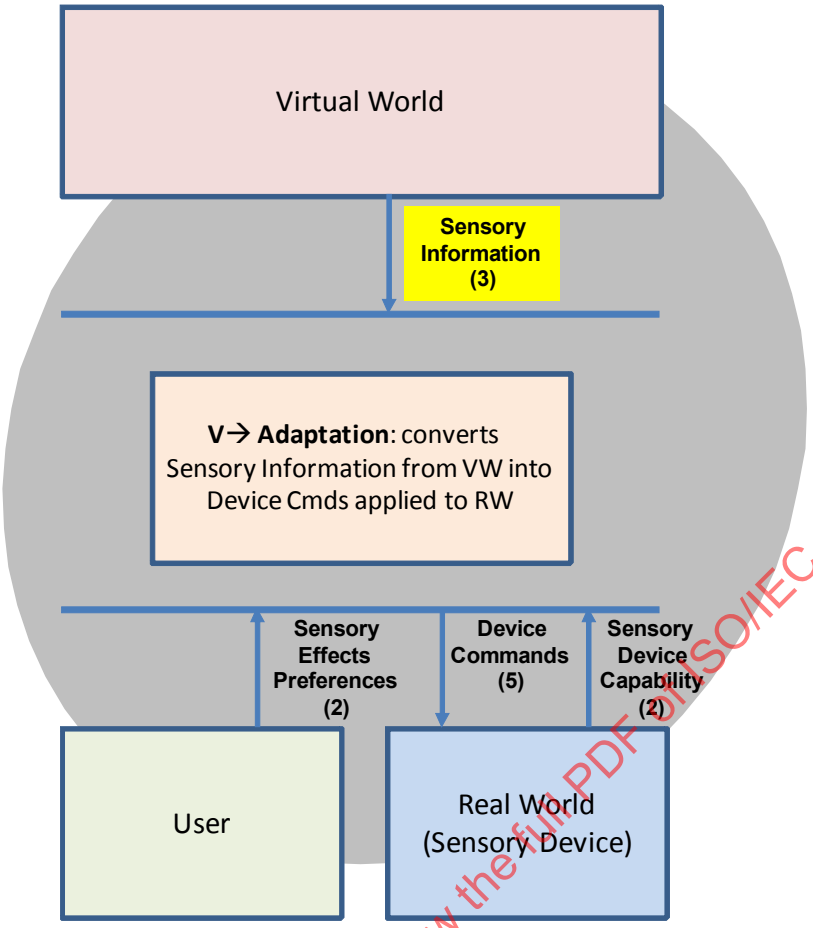


Figure 1 — System Architecture

NOTE 2 Additional informative information can be found in Annex A.

The usage scenarios are described in detail in MPEG-V Architecture (ISO/IEC 23005-1).

2 Normative references

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

ISO/IEC 21000-7, *Information technology — Multimedia framework (MPEG-21) — Part 7: Digital Item Adaptation*

ISO/IEC 23005 (all parts), *Information technology — Media context and control*

W3C XML, *Extensible Markup Language (XML) 1.1*, Second Edition, W3C Recommendation 16 August 2006, edited in place 29 September 2006

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

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

For the purpose of this document, the terms and definitions in ISO/IEC 23005-6 and the following apply.

3.1.1
digital content provider
entity that acts as the source of digital information of various nature

NOTE The digital content may be provided in real-time or non real-time.

EXAMPLE Digital content from an on-line virtual world, simulation environment, multi user game, a broadcasted multimedia production, a peer-to-peer multimedia production, or packaged content like a DVD or game.

3.2.2
sensory information
standardized representation format of ISO/IEC 23005 in the standardization area B as defined in ISO/IEC 23005-1

EXAMPLE Sensory effect metadata, haptic (kinesthetic/tactile) information, emotion information, avatar information.

3.2.3
sensory effect metadata
defines the description schemes and descriptors to represent **sensory effects**

3.2.4
sensory effect
effect to augment perception by stimulating human senses in a particular scene of a multimedia application

EXAMPLE Scent, wind, light, haptic(kinesthetic-force, stiffness, weight, friction, texture, widget (button, slider, joystick), tactile: air-jet, suction pressure, thermal, current, vibration, note: combinations of tactile display may provide also directional, shape information).

3.2.5
adaptation VR
entity that can process the **sensory information** in order to be consumed within the real world's context

NOTE This may include the adaptation or transformation of the sensory information according to the capabilities of real world devices or the preferences of the user. A specification of these capabilities and preferences can be found in ISO/IEC 23005-2.

3.2 Abbreviated terms

For the purpose of this document, the following abbreviated terms apply:

DIA digital item adaptation (ISO/IEC 21000-7)

MPEG-21 multimedia framework (ISO/EC 21000)

MPEG-7 multimedia content description interface (ISO/IEC 15938)

SEDL sensory effects description language

SEM sensory effect metadata

SEV sensory effects vocabulary

UMA universal multimedia access

UME universal multimedia experience

XML extensible mark-up language

XSI XML streaming instructions

3.3 Schema documents

In the main text of this specification, the syntax of description schemes and descriptors is provided whenever possible as a single schema document.

In some cases though, and in particular for Clause 5, the syntax of description schemes and descriptors is provided as a collection of schema snippets imbricated with other text. In order to form a valid schema document, these schema components should be gathered in a same document with the schema wrapper provided at the head of the clause. For better readability, the relevant schema documents are provided in Annex B, but as non-normative information.

In all cases, each schema document has a `version` attribute, the value of which is "ISO/IEC 23005-3". Furthermore, an informative identifier is given as the value of the `id` attribute of the `schema` component. This identifier is non-normative and used as a convention in this specification to reference another schema document. In particular, it is used for the `schemaLocation` attribute of the `include` and `import` schema components.

3.4 Use of prefixes

For clarity, throughout this Part of ISO/IEC 23005, consistent namespace prefixes are used.

"`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.

"`xml:`" and "`xmlns:`" are normative prefixes defined in [1]. The prefix "`xml:`" is by definition bound to "`http://www.w3.org/XML/1998/namespace`". The prefix "`xmlns:`" is used only for namespace bindings and is not itself bound to any namespace name.

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

In particular, most of the informative examples in this specification are provided as XML fragments without the normally required XML document declaration and, thus, miss a correct namespace binding context declaration. In these descriptions fragments the different prefixes are bound to the namespaces as given in the following table.

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

Prefix	Corresponding namespace
`ct`	`urn:mpeg:mpeg-v:2010:01-CT-NS`
`sedl`	`urn:mpeg:mpeg-v:2010:01-SEDL-NS`
`sev`	`urn:mpeg:mpeg-v:2010:01-SEV-NS`
`dia`	`urn:mpeg:mpeg21:2003:01-DIA-NS`
`si`	`urn:mpeg:mpeg21:2003:01-DIA-XSI-NS`
`mpeg7`	`urn:mpeg:mpeg7:schema:2004`
`xsi`	`http://www.w3.org/2001/XMLSchema-instance`
`xsd`	`http://www.w3.org/2001/XMLSchema`

Unlike the informative descriptions examples, the normative specification of the syntax of tools in XML Schema follows the namespace binding context defined in the relevant schema declaration such as the one defined in 5.3.

4 Sensory effects description language

4.1 Introduction

This Clause specifies the syntax and semantics of the sensory effects description language (SEDL) which provides basic building blocks for the authoring of sensory effect metadata.

4.2 Validation

Validating a document against the SEDL schema (as specified in W3C XMLSCHEMA) is necessary, but not sufficient, to determine its validity with respect to SEDL. After a document is validated against the SEDL 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.

4.3 Processing

The processing model for the sensory effect metadata is defined as an XML processor (as specified by W3C XML) and the utilization of the elements and attributes as defined in the subsequent (Sub)clauses.

NOTE The processing of the sensory effect metadata may follow existing XML decoding/parsing models such as the Document Object Model (DOM) or the Simple API for XML (SAX).

The time information that may be associated to sensory effects may be used for the synchronization with respect to other media assets.

EXAMPLE These other media assets may be video and/or audio.

4.4 Basic building blocks

4.4.1 Introduction

This Subclause specifies the syntax and semantics of the basic building blocks for authoring sensory effect metadata.

This Part of ISO/IEC 23005 adopts the XML streaming instructions (XSI) as defined in ISO/IEC 21000-7 for the purpose of identifying process units and associating time information to them. In this context, a process unit is defined as a well-formed fragment of XML-based metadata that can be consumed as such and to which time information may be attached, indicating the point in time when it becomes available for consumption. A process unit is specified by one element named anchor element and by a process unit mode indicating how other connected elements are aggregated to this anchor to compose the process unit. Depending on the mode, the anchor element is not necessarily the root of the process unit. Anchor elements are ordered according to the navigation path of the XML document. Process units may overlap, i.e. some elements (including anchor elements) may belong to several process units. Additionally, the content provider may require that a given process unit be encoded as a random access point, i.e. that the resulting access unit does not require any other access units to be decoded. The syntax and semantics of the XML streaming instructions is fully specified in 8.6 of ISO/IEC 21000-7:2007.

In addition to the XML streaming instructions, this standard adopts the following basic time model for sensory effects metadata which is depicted in Figure 2.

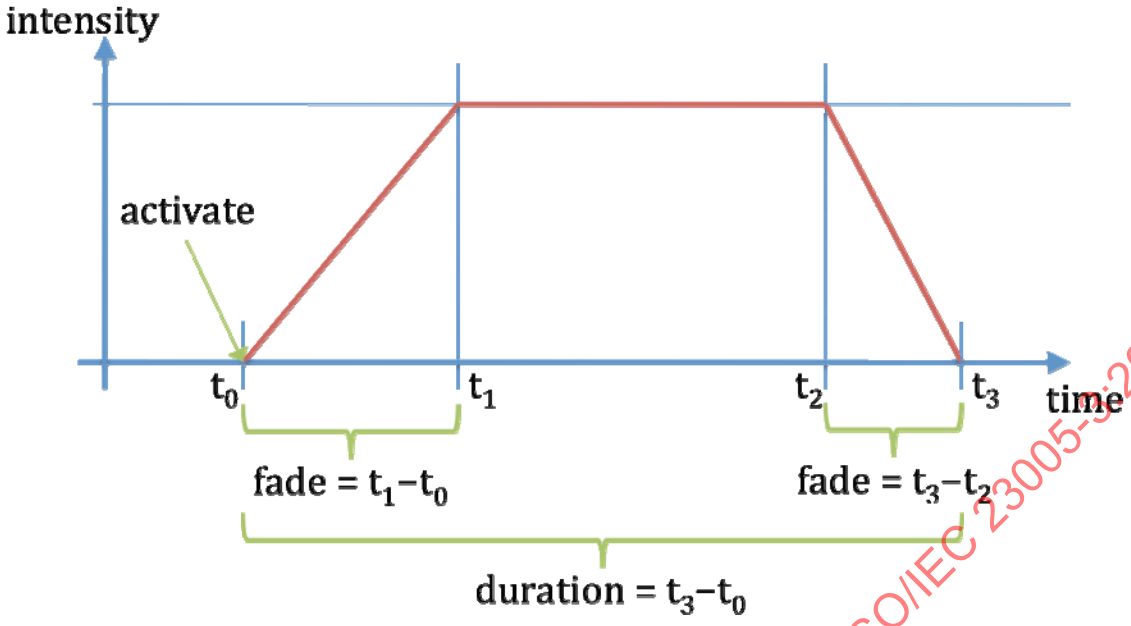


Figure 2 — Time model for sensory effect metadata

Each effect may be activated (i.e., t_0) and deactivated (i.e., t_3) at certain points in time. The deactivation of an effect may be explicitly defined (i.e., `activate="false"`) or indicated by means of a duration attribute during activation (i.e., t_3-t_0). Furthermore, each effect may specify a fade-in (i.e., t_1-t_0) or fade-out (i.e., t_3-t_2) time within which the corresponding effect shall reach its specified intensity.

NOTE The actual implementation of some effects may require one or more elements as defined in the following. An example implementation of Figure 2 using the syntax as defined in the following is provided in 4.4.14.

4.4.2 Schema wrapper

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

```
<?xml version="1.0"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sedl="urn:mpeg:mpeg-v:2010:01-SEDL-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
  xmlns:si="urn:mpeg:mpeg21:2003:01-DIA-XSI-NS"
  xmlns:ct="urn:mpeg:mpeg-v:2010:01-CT-NS"
  targetNamespace="urn:mpeg:mpeg-v:2010:01-SEDL-NS"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="ISO/IEC 23005-3" id="MPEG-V-SEDL.xsd">

  <import namespace="urn:mpeg:mpeg7:schema:2004"
schemaLocation="http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-
7_schema_files/mpeg7-v2.xsd"/>
  <import namespace="urn:mpeg:mpeg21:2003:01-DIA-XSI-NS"
schemaLocation="http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-
21_schema_files/dia-2nd/XSI-2nd.xsd"/>
  <import namespace="urn:mpeg:mpeg-v:2010:01-CT-NS" schemaLocation="MPEG-V-
CT.xsd"/>
```

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

```
</schema>
```

4.4.3 Mnemonics for binary representations

The mnemonics are defined in 4.2 of ISO/IEC 23005-6.

4.4.4 Common header for binary representations

The common header is defined in 4.3 of ISO/IEC 23005-6.

4.4.5 Base datatypes and elements

4.4.5.1 Syntax

```
<!-- ################################################ -->
<!--  SEM Base Attributes                             -->
<!-- ################################################ -->
<attributeGroup name="SEMBaseAttributes">
  <attribute name="activate" type="boolean" use="optional" />
  <attribute name="duration" type="positiveInteger" use="optional" />
  <attribute name="fade" type="positiveInteger" use="optional" />
  <attribute name="alt" type="anyURI" use="optional" />
  <attribute name="priority" type="positiveInteger" use="optional" />
  <attribute name="location" type="mpeg7:termReferenceType"
     use="optional"/>
  <attributeGroup ref="sedl:SEMAdaptabilityAttributes"/>
</attributeGroup>
<simpleType name="intensityValueType">
  <restriction base="float"/>
</simpleType>
<simpleType name="intensityRangeType">
  <restriction>
    <simpleType>
      <list itemType="float"/>
    </simpleType>
    <length value="2" fixed="true"/>
  </restriction>
</simpleType>

<!-- ################################################ -->
<!--  SEM Adaptability Attributes                     -->
<!-- ################################################ -->
<attributeGroup name="SEMAdaptabilityAttributes">
  <attribute name="adaptType" type="sedl:adaptTypeType" use="optional"/>
  <attribute name="adaptRange" type="sedl:adaptRangeType" default="10"
             use="optional"/>
</attributeGroup>
<simpleType name="adaptTypeType">
  <restriction base="NMTOKEN">
    <enumeration value="strict"/>
    <enumeration value="under"/>
```

```
      <enumeration value="over"/>
      <enumeration value="both"/>
    </restriction>
  </simpleType>
  <simpleType name="adaptRangeType">
    <restriction base="unsignedInt">
      <minInclusive value="0"/>
      <maxInclusive value="100"/>
    </restriction>
  </simpleType>

  <!-- ################################################ -->
  <!--  SEM Base Elements                               -->
  <!-- ################################################ -->
  <element name="Declarations" type="sedl:DeclarationsType" />
  <element name="GroupOfEffects" type="sedl:GroupOfEffectsType" />
  <element name="Effect" type="sedl:EffectBaseType" />
  <element name="ReferenceEffect" type="sedl:ReferenceEffectType" />
  <element name="Parameter" type="sedl:ParameterBaseType"/>

  <!-- ################################################ -->
  <!--  SEM Base type                                   -->
  <!-- ################################################ -->
  <complexType name="SEMBaseType" abstract="true">
    <complexContent>
      <restriction base="anyType">
        <attribute name="id" type="ID" use="optional"/>
      </restriction>
    </complexContent>
  </complexType>
```

4.4.5.2 Binary Representation

SEMBaseAttributes{	*Number of Bits*	*Mnemonic*
activateFlag	1	bslbf
durationFlag	1	bslbf
fadeFlag	1	bslbf
altFlag	1	bslbf
priorityFlag	1	bslbf
locationFlag	1	bslbf
if(activateFlag) {		
activate	1	bslbf
}		
if(durationFlag) {		
duration	32	uimsbf
}		
if(fadeFlag) {		
fade	32	uimsbf
}		
if(altFlag) {		
alt	See ISO 10646	UTF-8
}		
if(priorityFlag) {		
priority	32	uimsbf
}		
if(locationFlag) {		
location	7	bslbf

}		
SEMAdaptabilityAttributes		SEMAdaptabilityAttributes
}		

SEMAdaptabilityAttributes {	*Number of Bits*	*Mnemonic*
adaptTypeFlag	1	bslbf
adaptRangeFlag	1	bslbf
if(adaptTypeFlag){		
adaptType	2	bslbf (Table 2)
}		
if(adaptRangeFlag){		
adaptRange	7	bslbf
}		
}		

Declarations{	*Number of bits*	*Mnemonic*
DeclarationsType		DeclarationsType
}		

GroupOfEffects{	*Number of bits*	*Mnemonic*
GroupOfEffectsType		GroupOfEffectsType
}		

Effect{	*Number of bits*	*Mnemonic*
EffecetBaseType		EffectBaseType
}		

ReferenceEffect{	*Number of bits*	*Mnemonic*
ReferenceEffectType		RefernceEffectType
}		

Parameter{	*Number of bits*	*Mnemonic*
ParameterBaseType		ParameterBaseType
}		

SEMBaseType{	*Number of bits*	*Mnemonic*
idFlag	1	bslbf
if(idFlag) {		
id	See ISO 10646	UTF-8
}		
}		

4.4.5.3 Semantics

Semantics of the `SEMBaseAttributes`:

Name	*Definition*
`activateFlag`	This field, which is only present in the binary representation, indicates the presence of active attribute. If it is set to “1”, the active attribute is following.
`activate`	Describes whether the effect shall be activated. A value of `true` means the effect shall be activated and `false` means the effect shall be deactivated.
`durationFlag`	This field, which is only present in the binary representation, indicates the presence of duration attribute. If it is set to “1”, the duration attribute is following.
`duration`	Describes the duration according to the time scheme used. The time scheme used shall be identified by means of the `si:absTimeScheme` and `si:timeScale` attributes respectively.
`fadeFlag`	This field, which is only present in the binary representation, indicates the presence of fade attribute. If it is set to “1”, the fade attribute is following.
`fade`	Describes the fade time according to the time scheme used within which the defined `intensity` shall be reached. The time scheme used shall be identified by means of the `si:absTimeScheme` and `si:timeScale` attributes respectively.
`altFlag`	This field, which is only present in the binary representation, indicates the presence of alt attribute. If it is set to “1”, the alt attribute is following.
`alt`	Describes an alternative effect identified by URI. NOTE 1 The alternative might point to an effect – or list of effects – within the same description or an external description. NOTE 2 The alternative might be used in case the original effect cannot be processed. EXAMPLE 1 The alternative effect is chosen because the original intended effect cannot be processed due to lack of devices supporting this effect.
`priorityFlag`	This field, which is only present in the binary representation, indicates the presense of priorityFlag attribute. If it is set to “1”, the priotiry attribute is following.
`priority`	Describes the priority for effects with respect to other effects in the same group of effects sharing the same point in time when they should become available for consumption. A value of one indicates the highest priority and larger values indicate lower priorities. NOTE 3 The priority might by used to process effects – defined within a group of effects – according to the capabilities of the adaptation VR. EXAMPLE 2 The adaptation VR processes the individual effects of a group of effects according to their priority in descending order due to its limited capabilities. That is, effects with low priority might get lost.

Name	*Definition*
`locationFlag`	This field, which is only present in the binary representation, indicates the presence of location attribute. If it is set to “1”, the location attribute is following.
`location`	Describes the location from where the effect is expected to be received from the user’s perspective according to the x-, y-, and z-axis as depicted in Figure 3.

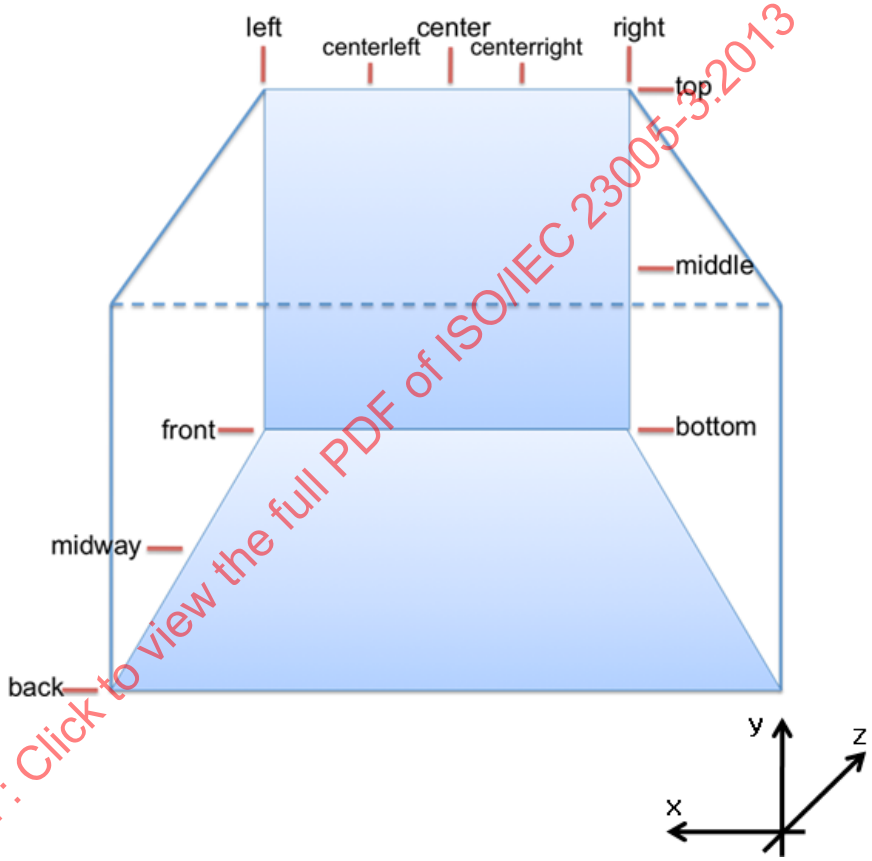


Figure 3 — Location model for sensory effect metadata and reference coordinate system

A classification scheme that may be used for this purpose is the `LocationCS` as defined in Annex A of ISO/IEC 23005-6. The terms from the `LocationCS` shall be concatenated with the “`:`” sign in order of the x-, y-, and z-axis to uniquely define a location within the three-dimensional space.

For referring to a group of locations, a wild card mechanism may be employed using the "`*`" sign.

EXAMPLE 4 `urn:mpeg:mpeg-v:01-SI-LocationCS-NS:center:middle:front` defines the location as follows: center on the x-axis, middle on the y-axis, and front on the z-axis. That is, it describes all effects at the center, middle, front side of the user.

EXAMPLE 5 `urn:mpeg:mpeg-v:01-SI-LocationCS-NS:left:*:midway` defines the location as follows: left on the x-axis, any location on the y-axis, and midway on the z-axis. That is, it describes all effects at the left, midway side of the user.

Name	*Definition*
	EXAMPLE 6 `urn:mpeg:mpeg-v:01-SI-LocationCS-NS:*:*:back` defines the location as follows: any location on the x-axis, any location on the y-axis, and back on the z-axis. That is, it describes all effects at the back of the user. The binary representation for the LocationCS is defined in Annex A of ISO/IEC 23005-6
`SEMAdaptabilityAttributes`	This field, which is only present in the binary representation, describes a group of attributes for the adaptability attributes of effects.

Semantics of the `intensityValueType` and `intensityRangeType`:

Name	*Definition*
`intensityValueType`	Tool for describing the intensity of the effect. Each effect must define its intensity value using this datatype.
`intensityRangeType`	Tool for describing the intensity range of the effect. Each effect must define its intensity range using this datatype.

Semantics of the `SEMAdaptabilityAttributes`:

Name	*Definition*
`adaptType`	Describes the preferred type of adaptation with the following possible instantiations: — `strict`: An adaptation by approximation may not be performed. — `under`: An adaptation by approximation may be performed with a smaller effect value than the specified effect value. NOTE 1 (1 – adaptRange) x intensity ~ intensity. — `over`: An adaptation by approximation may be performed with a greater effect value than the specified effect value. NOTE 2 intensity ~ (1 + adaptRange) x intensity. — `both`: An adaptation by approximation may be performed between the upper and lower bound specified by `adaptRange`. NOTE 3 (1 – adaptRange) x intensity ~ (1 + adaptRange) x intensity. In the binary description, the following mapping table is used, **Table 2 — adaptType** **adaptType** / **Sementics** 00 / Strict 01 / Under 10 / Over 11 / Both

Name	*Definition*
adaptTypeFlag	This field, which is only present in the binary representation, indicates the presence of adaptType attribute. If it is set to "1", the adaptType attribute is following.
adaptRange	Describes the upper and lower bound in percentage for the adaptType. If the adaptType is not present, adaptRange shall be ignored.

Semantics of the SEM base elements:

Name	*Definition*
Declarations	Describes a declaration of sensory effects, group of sensory effects, or parameters. NOTE 1 The declarations may be used by reference using the ReferenceEffect element.
Effect	Describes a sensory effect.
GroupOfEffects	Describes a group of sensory effects. NOTE 2 The purpose of grouping is to remove some redundancy from its child elements. All attributes included here are inherited to its child elements.
ReferenceEffect	Describes a reference to a sensory effect, group of sensory effects, or parameter. NOTE 3 The reference may point to a sensory effect, group of sensory effects, or parameter as defined within the same description or an external description by means of the Declarations element.
Parameter	Describes a parameter for a sensory effect. NOTE 4 The parameter may be used to declare complex properties to be used within sensory effects. As such, it shall be defined within the Declarations element.

Semantics of the SEMBaseType:

Name	*Definition*
SEMBaseType	Provides the topmost type of the base type hierarchy.
idFlag	This field, which is only present in the binary representation, indicates the presence of id attribute. If it is set to "1", the id attribute is following.
id	Identifies the id of the SEMBaseType.

4.4.6 Root element

4.4.6.1 Syntax

```
<!-- ################################################ -->
<!--  Definition of the SEM root element              -->
<!-- ################################################ -->
<element name="SEM">
  <complexType>
    <sequence>
      <element name="DescriptionMetadata"
           type="sedl:DescriptionMetadataType"
           minOccurs="0" maxOccurs="1"/>
      <choice maxOccurs="unbounded">
        <element ref="sedl:Declarations" />
        <element ref="sedl:GroupOfEffects" />
        <element ref="sedl:Effect" />
        <element ref="sedl:ReferenceEffect" />
      </choice>
    </sequence>
          <anyAttribute namespace="##other" processContents="lax"/>
  </complexType>
</element>
```

4.4.6.2 Binary representation syntax

SEM {	*Number of bits*	*Mnemonic*
DescriptionMetadataFlag	1	bslbf
If(DescriptionMetadataFlag){		
DescriptionMetadata		DescriptionMetadata
}		
NumOfElements		vluimsbf5
For(k=0;k<NumOfElements;k++){		
ElementID	4	uimsbf (Table 3)
Element		Element
}		
anyAttributeType		anyAttributeType
}		

anyAttributeType {	*Number of bits*	*Mnemonic*
siAttibutes		siAttributeList
anyAttributeFlag	1	bslbf
If(anyAttributeFlag) {		
SizeOfanyAttribute		vluimsbf5
anyAttribute	SizeOfanyAttribute *8	bslbf
}		
}		

siAttributeList {	*Number of bits*	*Mnemonic*
anchorElementFlag	1	bslbf
encodeAsRAPFlag	1	bslbf
puModeFlag	1	bslbf
timeScaleFlag	1	bslbf
ptsDeltaFlag	1	bslbf
absTimeSchemeFlag	1	bslbf
absTimeFlag	1	bslbf
ptsFlag	1	bslbf
absTimeSchemeLength		vluimsbf5
absTimeLength		vluimsbf5
if(anchorElementFlag) {		
anchorElement	1	bslbf
}		
if(encodeAsRAPFlag) {		
encodeAsRAP	1	bslbf
}		
if(puModeFlag) {		
puMode	3	bslbf (Table 4)
}		
if(puModeFlag) {		
timeScale	32	uimsbf
}		
if(ptsDeltaFlag) {		
ptsDelta	32	uimsbf
}		
if(absTimeSchemeFlag) {		
absTimeScheme	8*absTimeSchemeLength	bslbf
}		
if(absTimeFlag) {		
absTime	8*absTimeLength	bslbf
}		
if(ptsFlag) {		
pts		vluimsbf5
}		

4.4.6.3 Semantics

Semantics of the `SEM` root element:

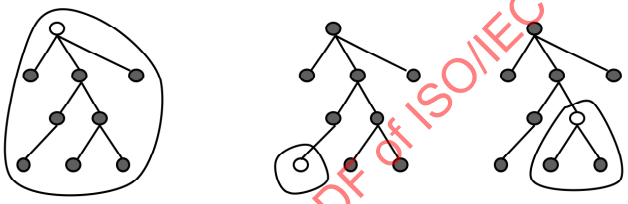
Name	*Definition*
`SEM`	Serves as the root element for sensory effects metadata.
`DescriptionMetadataFlag`	This field, which is only present in the binary representation, indicates the presence of the `DescriptionMetadata` element. If it is 1 then the `DescriptionMetadata` element is present, otherwise the `DescriptionMetadata` element is not present.
`DescriptionMetadata`	Describes general information about the sensory effects metadata. EXAMPLE 1 Creation information or Classification Scheme Alias.
`NumOfElements`	This field, which is only present in the binary representation, specifies the number of Element instances accommodated in the SEM.
`ElementID`	This field, which is only present in the binary representation, describes which SEM scheme shall be used. In the binary description, the following mapping table is used, **Table 3 — Element ID** **ElementID** / **Element** 0 / Reserved 1 / Declarations 2 / GroupOfEffects 3 / Effect 4 / ReferenceEffect 5 / Parameter 6~15 / Reserved
`Element`	See 4.4.8, 4.4.9, 4.4.10, 4.4.11, and 4.4.12.
`Declarations`	See semantics of the SEM base elements.
`Effect`	See semantics of the SEM base elements.
`GroupOfEffects`	See semantics of the SEM base elements.
`ReferenceEffect`	See semantics of the SEM base elements.
`anyAttribute`	Provides an extension mechanism for including attributes from namespaces other than the target namespace. Attributes that shall be included are the XML streaming instructions as defined in ISO/IEC 21000-7 for the purpose of identifying process units and associating time information to them. EXAMPLE 2 `si:timeScale` describes the time scale to be used.

Name	*Definition*
anyAttributeType	Type of anyAttribure
siAttibutes	Make reference to follow siAttributeList
anyAttributeFlag	This field, which is only present in the binary representation, indicates the presence of anyAttribute attribute. If it is set to "1", the anyAttribute is following.
SizeOfanyAttribute	This field, which is only present in the binary representation, indicates the number of byte arrary for anyAttribute

Semantics of the siAttributeList:

Names	Description
anchorElementFlag	This field, which is only present in the binary representation, indicates the presence of the anchorElement attribute. If it is 1 then the anchorElement attribute is present, otherwise the anchorElement attribute is not present.
encodeAsRAPFlag	This field, which is only present in the binary representation, indicates the presence of the encodeAsRAP attribute. If it is 1 then the encodeAsRAP attribute is present, otherwise the encodeAsRAP attribute is not present.
puModeFlag	This field, which is only present in the binary representation, indicates the presence of the puMode attribute. If it is 1 then the puMode attribute is present, otherwise the puMode attribute is not present.
timeScaleFlag	This field, which is only present in the binary representation, indicates the presence of the timeScale attribute. If it is 1 then the timeScale attribute is present, otherwise the timeScale attribute is not present.
ptsDeltaFlag	This field, which is only present in the binary representation, indicates the presence of the ptsDelta attribute. If it is 1 then the ptsDelta attribute is present, otherwise the ptsDelta attribute is not present.
absTimeSchemeFlag	This field, which is only present in the binary representation, indicates the presence of the activation attribute. If it is 1 then the activation attribute is present, otherwise the activation attribute is not present.
absTimeFlag	This field, which is only present in the binary representation, indicates the presence of the absTimeScheme attribute. If it is 1 then the absTimeScheme attribute is present, otherwise the absTimeScheme attribute is not present.
ptsFlag	This field, which is only present in the binary representation, indicates the presence of the pts attribute. If it is 1 then the pts attribute is present, otherwise the pts attribute is not present.
absTimeSchemeLength	This field, which is only present in the binary representation, specifies the length of each absTimeSchemeLength instance in bytes. The value of this element is the size of the largest absTimeSchemeLength instance, aligned to a byte boundary by bit stuffing using 0-7 ' 1' bits.
absTimeLength	This field, which is only present in the binary representation, specifies the length of each absTimeLength instance in bytes. The value of this element is the size of the largest absTimeLength instance, aligned to a byte boundary by bit stuffing using 0-7 ' 1' bits.

<table>
<tr><td>anchorElement</td><td>Describes whether the element shall be anchor element. A value of <code>true(=1)</code> means the element shall be anchor element and <code>false(=0)</code> means the element shall be not anchor element.
The <code>anchorElement</code> allows one to indicate whether an XML element is an anchor element, i.e., the starting point for composing the process unit.</td></tr>
<tr><td>encodeAsRAP</td><td>Describes property indicates that the process unit shall be encoded as a random access point. A value of <code>true(=1)</code> means the process unit shall be encoded as a random access point and <code>false(=0)</code> means the process unit shall be not encoded as a random access point.</td></tr>
<tr><td>puMode</td><td>The <code>puMode</code> specifies how elements are aggregated to the anchor element to compose the process unit. For detailed information the reader is referred to ISO/IEC 21000-7 XSI.

Figure 4 shows an example for <code>puMode = descendants</code> which means that the process unit contains the anchor element and its descendant elements. Note that the anchor elements are pictured in white.
 
Figure 4 — Examples of process units - <code>puMode = descendants</code>

In the binary description, the following mapping table is used.

Table 4 — putMode
<table><tr><td>puMode</td><td>puModeType</td></tr><tr><td>000</td><td>self</td></tr><tr><td>001</td><td>ancestors</td></tr><tr><td>010</td><td>descendants</td></tr><tr><td>011</td><td>ancestorsDescendants</td></tr><tr><td>100</td><td>preceding</td></tr><tr><td>101</td><td>precedingSiblings</td></tr><tr><td>110</td><td>sequential</td></tr><tr><td>111</td><td>Reserved</td></tr></table></td></tr>
<tr><td>timeScale</td><td>Describes a time scale.</td></tr>
<tr><td>ptsDelta</td><td>Describes a processing time stamp delta.</td></tr>
<tr><td>absTimeScheme</td><td>Describes an absolute time scheme.</td></tr>
<tr><td>absTime</td><td>Describes an absolute time.</td></tr>
<tr><td>pts</td><td>Describes a processing time stamp (PTS).</td></tr>
</table>

4.4.7 Description metadata

4.4.7.1 Syntax

```
<!-- ################################################ -->
<!--  Definition of Description Metadata Type         -->
<!-- ################################################ -->
<complexType name="DescriptionMetadataType">
  <complexContent>
    <extension base="mpeg7:DescriptionMetadataType">
```

```
        <sequence>
          <element name="ClassificationSchemeAlias" minOccurs="0"
             maxOccurs="unbounded">
            <complexType>
              <complexContent>
                <extension base="sedl:SEMBaseType">
                  <attribute name="alias" type="NMTOKEN" use="required"/>
                  <attribute name="href" type="anyURI" use="required"/>
                </extension>
              </complexContent>
            </complexType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

4.4.7.2 Binary representation syntax

DescriptionMetadata{	*Number of bits*	*Mnemonic*
MPEG7DescriptionMetadata		Mpeg7:DescriptionMetadataType
NumOfClassSchemeAlias		vluimsbf5
for(k=0;k<NumOfClassSchemeAlias;k++){		
SEMBaseType[k]		SEMBaseType
alias[k]	See ISO 10646	UTF-8
href[k]		UTF-8
}		
}		

4.4.7.3 Semantics

Semantics of the `DescriptionMetadata`:

Name	*Definition*
`DescriptionMetadataType`	`DescriptionMetadataType` extends `mpeg7:DescriptionMetadataType` and provides a sequence of classification schemes for usage in the `SEM` description.
`MPEG7DescriptionMetadata`	make reference to `MPEG7:DescriptionMetadata`
`NumOfClassSchemeAlias`	This field, which is only present in the binary representation, specifies the number of Classification Scheme Alias instances accommodated in the `description metadata`.
`ClassificationSchemeAlias`	Describes an `alias` for a classification scheme referenced by URI.
`SEMBase`	Describes a base type of a Sensory Effect Metadata.

Name	*Definition*
`alias`	Describes the alias assigned to the `ClassificationScheme`. The scope of the alias assigned shall be the entire description regardless of where the `ClassificationSchemeAlias` appears in the description
`href`	Describes a reference to the classification scheme that is being aliased using a URI. The classification schemes defined in this part of the ISO/IEC 23005, whether normative of informative, shall be referenced by the `uri` attribute of the `ClassificationScheme` for that classification scheme.

4.4.8 Declarations

4.4.8.1 Syntax

```
<!-- ################################################ -->
<!--  Declarations type                               -->
<!-- ################################################ -->
<complexType name="DeclarationsType">
  <complexContent>
    <extension base="sedl:SEMBaseType">
      <choice maxOccurs="unbounded">
        <element ref="sedl:GroupOfEffects" />
        <element ref="sedl:Effect" />
        <element ref="sedl:Parameter" />
      </choice>
    </extension>
  </complexContent>
</complexType>
```

4.4.8.2 Binary represenstation syntax

Declarations{	*Number of bits*	*Mnemonic*
SEMBaseType		SEMBaseType
NumOfElements		vluimsbf5
For(k=0;k<NumOfElements;k++){		
ElementID	4	bslbf
Element		Element
}		
}		

*ElementID restricted 2, 3, 5

4.4.8.3 Semantics

Semantics of the `DeclarationsType`:

Name	*Definition*
`DeclarationsType`	Tool for describing a declaration of sensory effects, group of sensory effects, or parameters. NOTE The declarations may be used by reference using the `ReferenceEffect` element.
`Effect`	See semantics of the SEM base elements.
`GroupOfEffects`	See semantics of the SEM base elements.
`Parameter`	See semantics of the SEM base elements.
`SEMBaseType`	Describes a base type of a Sensory Effect Metadata.
`NumOfElements`	This field, which is only present in the binary representation, specifies the number of Element instances accommodated in the SEM.
`ElementID`	This field, which is only present in the binary representation, describes which SEM scheme shall be used. In the binary description, make referece to Table 3. Element ID

4.4.9 Group of effects

4.4.9.1 Syntax

```
<!-- ################################################ -->
<!--  Group of Effects type                           -->
<!-- ################################################ -->
<complexType name="GroupOfEffectsType">
  <complexContent>
    <extension base="sedl:SEMBaseType">
      <choice minOccurs="2" maxOccurs="unbounded">
        <element ref="sedl:Effect"/>
        <element ref="sedl:ReferenceEffect"/>
      </choice>
      <attributeGroup ref="sedl:SEMBaseAttributes"/>
      <anyAttribute namespace="##other" processContents="lax"/>
    </extension>
  </complexContent>
</complexType>
```

4.4.9.2 Binary representation syntax

GroupOfEffects{	*Number of bits*	*Mnemonic*
SEMBaseType		SEMBaseType
NumOfElements	5	uimsbf

For(k=0;k<NumOfElements;k++){		
ElementID	4	bslbf
Element		bslbf
}		
SEMBaseAttributes		SEMBaseAttributes
anyAttributeType		anyAttributeType
}		

4.4.9.3 Semantics

Semantics of the `GroupOfEffectsType`:

Name	*Definition*
`GroupOfEffectsType`	Tool for describing a group of two or more sensory effects.
`Effect`	See semantics of the SEM base elements.
`NumOfElements`	This field, which is only present in the binary representation, specifies the number of Element instances accommodated in the SEM.
`ElementID`	This field, which is only present in the binary representation, describes which SEM scheme shall be used. In the binary description, make referece to Table 3. Element ID NOTE ElementID restricted 3, 4
`SEMBaseAttributes`	Describes a group of attributes for the group of effects.
`anyAttribute`	Provides an extension mechanism for including attributes from namespaces other than the target namespace. Attributes that shall be included are the XML streaming instructions as defined in ISO/IEC 21000-7 for the purpose of identifying process units and associating time information to them. EXAMPLE `si:pts` describes the point in time when the associated information shall become available to the application for processing.

4.4.10 Effect

4.4.10.1 Syntax

```
  <!-- ################################################ -->
  <!--  Effect base type                                -->
  <!-- ################################################ -->
  <complexType name="EffectBaseType" abstract="true">
    <complexContent>
      <extension base="sedl:SEMBaseType">
        <sequence minOccurs="0">
          <element name="SupplementalInformation"
type="sedl:SupplementalInformationType" minOccurs="0"/>
```

```
      </sequence>
      <attribute name="autoExtraction" type="sedl:autoExtractionType"/>
      <attributeGroup ref="sedl:SEMBaseAttributes"/>
      <anyAttribute namespace="##other" processContents="lax"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="SupplementalInformationType">
  <sequence>
    <element name="ReferenceRegion" type="mpeg7:SpatioTemporalLocatorType"/>
    <element name="Operator" type="sedl:OperatorType" minOccurs="0"/>
  </sequence>
</complexType>
<simpleType name="OperatorType">
  <restriction base="NMTOKEN">
    <enumeration value="average"/>
    <enumeration value="dominant"/>
  </restriction>
</simpleType>
<simpleType name="autoExtractionType">
  <restriction base="string">
    <enumeration value="audio"/>
    <enumeration value="visual"/>
    <enumeration value="both"/>
  </restriction>
</simpleType>
```

4.4.10.2 Binary representation syntax

Effect{	*Number of bits*	*Mnemonic*
EffectTypeID	8	uimsbf(Table 5)
EffectbaseType		EffectbaseType
EffectType		EffectType
}		

EffectBaseType{	*Number of bits*	*Mnemonic*
SEMBaseType		SEMBaseType
SupplementalInformationType		SupplementalInformationType
autoExtractionID	2	uimsbf (Table 7)
SEMBaseAttributes		SEMBaseAttributes
anyAttributeType		anyAttributeType
}		

SupplementalInformationType {	*Number of bits*	*Mnemonic*
ReferenceRegion		ISO/IEC 15938-3:2002(E)
Operator	3	bslbf (Table 6)
}		

SpatioTemporalLocatorType {	*Number of bits*	*Mnemonic*
CoordFlag	1	bslbf
If(CoordFlag)		
refLength		vluimsbf5
ref	8*refLength	bslbf
spatialRef	1	bslbf
}		
NumOfRefRegions		vluimsbf5
for(k=0;k<NumOfRefRegions;k++){		
TypeOfTrajectory	2	bslbf
if(TypeOfTrajectory==”00”){		
FigureTrajectory		FigureTrajectoryType
}else if(TypeOfTrajectory==”01”){		
ParameterTrajectory		ParameterTrajectoryType
}else if(TypeOfTrajectory==”10”){		
MediaTime		MediaTimeType
}		
}		
}		

4.4.10.3 Semantics

Semantics of the `EffectBaseType`:

Name	*Definition*
`EffectTypeID`	This field, which is only present in the binary representation, specifies a descriptor identifier. The descriptor identifier indicates the descriptor type accommodated in the Effect.
`EffectBaseType`	`EffectBaseType` extends `SEMBaseType` and provides a base abstract type for a subset of types defined as part of the sensory effects metadata types.

Table 5 — EffectType ID

EffectType ID	**EffectType**
0	Reserved
1	LightType
2	FlashType
3	TemperatureType
4	WindType
5	VibrationType
6	SprayingType
8	FogType
9	ColorCorrectionType
10	RigidBodyMotionType
11	PassiveKinesthetic MotionType
12	PassiveKinesthetic ForceType
13	ActiveKinestheticType
14	TactileType
15	ParameterizedTactileType
16	Reserved

Name	*Definition*
SEMBaseAttributes	Describes a group of attributes for the effects.
anyAttribute	Provides an extension mechanism for including attributes from namespaces other than the target namespace. Attributes that shall be included are the XML streaming instructions as defined in ISO/IEC 21000-7 for the purpose of identifying process units and associating time information to them. EXAMPLE si:pts describes the point in time when the associated information shall become available to the application for processing.

Semantics of the SupplementalInformationType:

Name	*Definition*
SupplementalInformationType	Tool for describing supplemental information.
ReferenceRegion	Describes the reference region for automatic extraction from video. If the autoExtraction is not present or is not equal to video, this element shall be ignored. The localization scheme used is identified by means of the mpeg7:SpatioTemporalLocatorType that is defined in ISO/IEC 15938-5.
Operator	Describes the preferred type of operator for extracting sensory effects from the reference region of video with the following possible instantiations. — average: extracts sensory effects from the reference region by calculating average value. · dominant: extracts sensory effects from the reference region by calculating dominant value. In the binary description, the following mapping table is used. **Table 6 — Operator** **Operator** | **Sementics** 000 | Reserved 001 | Average 010 | Dominant 011~111 | Reserved

Semantics of the `autoExtractionType`:

Name	*Definition*
`autoExtractionID`	This field, which is only present in the binary representation, indicates the media resource. In the binary description, the following mapping table is used. **Table 7 — autoExtractionID** **autoExtractionID** / **autoExtractionType** 00 / audio 01 / visual 10 / both 11 / Reserved
`autoExtraction`	Describes whether an automatic extraction of sensory effects from the media resource, which is described by this sensory effect metadata, is preferable (but not required). The following values are available: • `audio`: the automatic extraction of sensory effects from the audio part of the media resource, which is described by this sensory effect metadata, is preferable. EXAMPLE Audio autoExtraction Type which is based on audio dynamics and acoustic feature of the media resource, can extract Vibration, Spraying, Wind, etc. effect types • `visual`: the automatic extraction of sensory effects from the visual part of the media resource, which is described by this sensory effect metadata, is preferable. EXAMPLE Visual autoExtraction Type which is based on color, brightness, motion detection, can extract various sensory effect types such as Motion, Light, Flash, etc. • `both`: the automatic extraction of sensory effects from both the audio and visual part of the media resource, which is described by this sensory effect metadata, is preferable. If the autoextraction is on and the device is able to compute the effect, the result of the computation should overwrite the effect specified by the content designer.

4.4.11 Reference effect

4.4.11.1 Syntax

```
<!-- ################################################ -->
<!--  Reference Effect type                           -->
<!-- ################################################ -->
<complexType name="ReferenceEffectType">
  <complexContent>
    <extension base="sedl:SEMBaseType">
      <attribute name="uri" type="anyURI" use="required" />
      <attributeGroup ref="sedl:SEMBaseAttributes"/>
      <anyAttribute namespace="##other" processContents="lax" />
    </extension>
  </complexContent>
</complexType>
```

4.4.11.2 Binary representation syntax

ReferenceEffect{	*Number of bits*	*Mnemonic*
SEMBaseType		SEMBaseType
uri	See ISO 10646	UTF-8
SEMBaseAttributes		SEMBaseAttributes
anyAttributeType		anyAttributeType
}		

4.4.11.3 Semantics

Semantics of the `ReferenceEffectType`:

Name	*Definition*
`ReferenceEffectType`	Tool for describing a reference to a sensory effect, group of sensory effects, or parameter.
`uri`	Describes a reference to a sensory effect, group of sensory effects, or parameter by an Uniform Resource Identifier (URI). Its target type must be one – or derived – of `sedl:EffectBaseType`, `sedl:GroupOfEffectType`, or `sedl:ParameterBaseType`.
`SEMBaseAttributes`	Describes a group of attributes for the effects.
`anyAttribute`	Provides an extension mechanism for including attributes from namespaces other than the target namespace. Attributes that shall be included are the XML streaming instructions as defined in ISO/IEC 21000-7 for the purpose of identifying process units and associating time information to them. Attributes included here override the attribute values possibly defined within the sensory effect, group of effects or parameter referenced by the `uri`. EXAMPLE `si:pts` describes the point in time when the associated information shall become available to the application for processing.

4.4.12 Parameters

4.4.12.1 Syntax

```
<!-- ################################################ -->
<!--  Parameter Base type                             -->
<!-- ################################################ -->
<complexType name="ParameterBaseType" abstract="true">
  <complexContent>
    <extension base="sedl:SEMBaseType"/>
  </complexContent>
</complexType>

<!-- ################################################ -->
<!--  Definition of Color Correction Parameter type   -->
<!-- ################################################ -->
```

```
<complexType name="ColorCorrectionParameterType">
  <complexContent>
    <extension base="sedl:ParameterBaseType">
      <sequence>
        <element name="ToneReproductionCurves"
          type="ct:ToneReproductionCurvesType" minOccurs="0"/>
        <element name="ConversionLUT"
          type="ct:ConversionLUTType"/>
        <element name="ColorTemperature" type="ct:IlluminantType"
          minOccurs="0"/>
        <element name="InputDeviceColorGamut"
          type="ct:InputDeviceColorGamutType" minOccurs="0"/>
        <element name="IlluminanceOfSurround" type="mpeg7:unsigned12"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

4.4.12.2 Binary representation syntax

ParameterBaseType{	*Number of bits*	*Mnemonic*
SEMBaseType		SEMBaseType
}		

ColorCorrectionParameterType {	*Number of bits*	*Mnemonic*
ParameterBaseType		ParameterBaseType
ToneReproductionFlag	1	bslbf
ColorTemperatureFlag	1	bslbf
InputDeviceColorGamutFlag	1	bslbf
IlluminanceOfSurroundFlag	1	bslbf
if(ToneReproductionFlag) {		
ToneReproductionCurves		ToneReproductionCurvesType
}		
ConversionLUT		ConversionLUTType
if(ColorTemperatureFlag) {		
ColorTemperature		IlluminantType
}		
if(InputDeviceColorGamutFlag) {		
InputDeviceColorGamut		InputDeviceColorGamutType
}		
if(IlluminanceOfSurroundFlag) {		
IlluminanceOfSurround	12	uimsbf
}		
}		

4.4.12.3 Semantics

Semantics of the `ParameterBaseType`:

Name	*Definition*
`ParameterBaseType`	Provides the topmost type of the parameter base type hierarchy.

Semantics of the `ColorCorrectionParameterType`:

Name	*Definition*
`ColorCorrectionParameterType`	A type defining the schema of the color correction effect.
`ToneReproductionFlag`	This field, which is only present in the binary representation, indicates the presence of the `ToneReproductionCurves` element. If it is 1 then the `ToneReproductionCurves` element is present, otherwise the `ToneReproductionCurves` element is not present.
`ToneReproductionCurves`	This curve shows the characteristics (e.g., gamma curves for R, G and B channels) of the input display device.
`ConversionLUT`	A look-up table (matrix) converting an image between an image color space (e.g. RGB) and a standard connection space (e.g. CIE XYZ).
`ColorTemperatureFlag`	This field, which is only present in the binary representation, indicates the presence of the `ColorTemperature` element. If it is 1 then the `ColorTemperature` element is present, otherwise the `ColorTemperature` element is not present.
`ColorTemperature`	An element describing a white point setting (e.g., D65, D93) of the input display device.
`InputDeviceColorGamutFlag`	This field, which is only present in the binary representation, indicates the presence of the `InputDeviceColorGamut` element. If it is 1 then the `InputDeviceColorGamut` element is present, otherwise the `InputDeviceColorGamut` element is not present.
`InputDeviceColorGamut`	An element describing an input display device color gamut, which is represented by chromaticity values of R, G, and B channels at maximum DAC values.
`IlluminanceOfSurroundFlag`	This field, which is only present in the binary representation, indicates the presence of the `IlluminanceOfSurround` element. If it is 1 then the `IlluminanceOfSurround` element is present, otherwise the `IlluminanceOfSurround` element is not present.
`IlluminanceOfSurround`	An element describing an illuminance level of viewing environment. The illuminance is represented by lux.

4.4.13 Additional validation rules

4.4.13.1 Introduction

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

4.4.13.2 The `SEM` element shall have a `si:timeScale` attribute.

4.4.13.3 The following rules shall apply on the `GroupOfEffects`.

4.4.13.3.1 A `GroupOfEffects` shall have a timestamp (i.e., `pts`, `ptsDelta`, or `absTime`).

4.4.13.3.2 A `GroupOfEffects` outside of a `Declarations` shall not have both a `pts` and an `absTime` at the same time.

4.4.13.3.3 A `GroupOfEffects` within a `Declarations` shall have only a `ptsDelta` for a timestamp.

4.4.13.4 The following rules shall apply on the `Effect`.

4.4.13.4.1 At least `activate`, `duration`, or `fade` shall be defined.

4.4.13.4.2 An `Effect` outside of a `GroupOfEffects` shall have a timestamp (i.e., `pts`, `ptsDelta`, or `absTime`).

4.4.13.4.3 An `Effect` within a `GroupOfEffects` shall have only a `ptsDelta` for a timestamp.

4.4.13.4.4 An `Effect` shall not have both a `pts` and an `absTime` at the same time.

4.4.13.4.5 An `Effect` within a `Declarations` shall have only a `ptsDelta` for a timestamp.

4.4.13.4.6 If `duration` is defined `activate` may not be defined.

4.4.13.4.7 If `fade` and `duration` are defined `activate` may not be defined.

4.4.13.4.8 If `fade` is defined intensity shall be defined.

NOTE The actual intensity is defined within the individual effecs by defining its value and range as an extension of `intensityValueType` and `intensityRangeType` respectively.

EXAMPLE `intensity-value` and `intensity-range`.

4.4.13.4.9 If two (or more) consecutive `Effect` elements of the same type share the same timestamp (i.e., `pts`, `ptsDelta`, or `absTime`) and the same location or overlap in time (i.e., `location`, `duration`, `activate` set to `true`/`false` resulting in overlap) only the latest in their order of appearance shall be used while retaining the semantics of the `priority` attribute.

NOTE It is possible to remove the other `Effect` elements from the description.

4.4.13.4.10 If `fade` and `duration` are defined `fade` must be less or equal to `duration`.

4.4.13.5 The following rules shall apply on the `ReferenceEffect`.

4.4.13.5.1 A `ReferenceEffect` outside of a `GroupOfEffects` shall have a timestamp (i.e., `pts`, `ptsDelta`, or `absTime`).

4.4.13.5.2 A `ReferenceEffect` within a `GroupOfEffects` shall have only a `ptsDelta` for a timestamp.

4.4.13.5.3 A `ReferenceEffect` shall not have both a `pts` and an `absTime` at the same time.

4.4.13.5.4 A `ReferenceEffect` within `Declarations` shall have only a `ptsDelta` for a timestamp.

4.4.14 Examples

EXAMPLE 1 The following example shows a possible usage of the basic building blocks.

```
<SEM>
  <DescriptionMetadata>
    <ClassificationSchemeAlias alias="..." href="..."/>
  </DescriptionMetadata>

  <Declarations>
    <!-- some declarations to be used here -->
  </Declarations>

  <GroupOfEffects ...>
    <Effect .../>
    <Effect .../>
  </GroupOfEffects>
  <ReferenceEffect uri="#eff1" .../>
  <Effect .../>
  <GroupOfEffects ...>
    <Effect .../>
    <Effect .../>
    <Effect .../>
  </GroupOfEffects>
  <ReferenceEffect uri="#eff1" .../>
  <!-- and so on -->
</SEM>
```

EXAMPLE 2 The following example shows a possible abstract implementation of Figure 2. Note that the example is declared as abstract because the attribute values refer to variables introduced in Figure 2 instead of using real values.

```
<Effect si:pts="t0" activate="true" fade="t1-t0" intensity-value="int"/>
<Effect si:pts="t2" activate="false" fade="t3-t2" intensity-value="0"/>
```

5 Sensory effect vocabulary

5.1 Introduction

This Clause specifies syntax and semantics of the sensory effect vocabulary which comprises the following effects:

— Light, colored light, flash light;

— Temperature;

— Wind;

— Vibration;

— Water sprayer;

— Scent;

— Fog;

— Color correction;

— Rigid body motion;

— Passive kinesthetic motion;

— Passive kinesthetic force;

— Active kinesthetic;

— Tactile;

NOTE SEV has been designed in an extensible way and additional sensory effects can be added easily.

EXAMPLE Additional sensory effects may be added as extensions to `sedl:EffectBaseType` and conformance to SEDL.

5.2 Validation

Validating a document against the SEV schema (as specified in W3C XMLSCHEMA) is necessary, but not sufficient, to determine its validity with respect to SEV. After a document is validated against the SEV 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.

5.3 Schema wrapper

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

```
<?xml version="1.0"?>
<schema
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sedl="urn:mpeg:mpeg-v:2010:01-SEDL-NS"
  xmlns:sev="urn:mpeg:mpeg-v:2010:01-SEV-NS"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
  targetNamespace="urn:mpeg:mpeg-v:2010:01-SEV-NS"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="ISO/IEC 23005-3" id="MPEG-V-SEV.xsd">

  <import namespace="urn:mpeg:mpeg-v:2010:01-SEDL-NS" schemaLocation="MPEG-V-
SEDL.xsd"/>
<import namespace="urn:mpeg:mpeg7:schema:2004"
schemaLocation="http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-
7_schema_files/mpeg7-v2.xsd"/>
```

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

```
</schema>
```

5.4 Light effect

5.4.1 Introduction

This Subclause specifies syntax and semantics of a light effect.

5.4.2 Syntax

```
  <!-- ########################################### -->
  <!--  SEV Light type                             -->
  <!-- ########################################### -->
  <complexType name="LightType">
    <complexContent>
      <extension base="sedl:EffectBaseType">
        <attribute name="color" type="sev:colorType" use="optional"/>
        <attribute name="intensity-value" type="sedl:intensityValueType"
          use="optional"/>
        <attribute name="intensity-range" type="sedl:intensityRangeType"
          use="optional"/>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="colorType">
    <union memberTypes="mpeg7:termReferenceType sev:colorRGBType"/>
  </simpleType>

  <simpleType name="colorRGBType">
    <restriction base="NMTOKEN">
      <whiteSpace value="collapse"/>
      <pattern value="#[0-9A-Fa-f]{6}"/>
    </restriction>
  </simpleType>
```

5.4.3 Binary representation syntax

LightType{	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
colorFlag	1	bslbf
intensityValueFlag	1	bslbf
intensityRangeFlag	1	bslbf
if(colorFlag) {		
color	~~9~~	colorType
}		
if(intensityValueFlag) {		
intensityValue	32	fsfb
}		
if(intensityRangeFlag) {		
intensityRange[0]	32	fsfb
intensityRange[1]	32	fsfb
}		
}		

5.4.4 Semantics

Semantics of the `LightType`:

Name	*Definition*
`LightType`	Tool for describing a light effect.
`colorFlag`	This field, which is only present in the binary representation, indicates the presence of the `color` attribute. If it is 1 then the `color` attribute is present, otherwise the `color` attribute is not present.
`intensityValueFlag`	This field, which is only present in the binary representation, indicates the presence of the `intensity-value` attribute. If it is 1 then the `intensity-value` attribute is present, otherwise the `intensity-value` attribute is not present.
`intensityRangeFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityRange attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
`color`	Describes the color of the light effect as a reference to a classification scheme term or as RGB value. The binary representation is defined in ISO/IEC 23005-6. A CS that may be used for this purpose is the `ColorCS` defined in Annex A of ISO/IEC 23005-6. EXAMPLE 1 `urn:mpeg:mpeg-v:01-SI-ColorCS-NS:alice_blue` would describe the color Alice blue.

Name	*Definition*
`intensity-value`	Describes the intensity of the light effect in terms of illumination in lux.
`intensity-range`	Describes the range of the intensity value. EXAMPLE 2 [10.0^{-5} lux, 130.0 klx].

Semantics of the `ColorType`:

Name	*Definition*
`NamedcolorFlag`	This field, which is only present in the binary representation, indicates a choice of the color descriptions. If it is 1 then the color is described by mpeg7:termReferenceType, otherwise the color is described by colorRGBType.
`NamedColorType`	This field, which is only present in the binary representation, describes color in terms of ColorCS Flag in Annex A.2.2 of ISO/IEC 23005-6.
`colorRGBType`	Tool for describing a color as RGB. EXAMPLE 3 `#F0F8FF` would describe the color Alice blue.
`fade`	In addition to the semantics as defined in 4.4.4, the following semantics apply: — If `fade` and `intensity` is provided, the fading shall be performed to the intensity defined in the `intensity` attribute. NOTE This is basically covered by the semantics as defined in 4.3.3.2 but repeated here fore better readability. — If `fade` and `color` is provided, the fading shall be performed to the color defined in the `color` attribute. — If `fade`, `intensity`, and `color` is provided, the fading shall be performed to the intensity and color defined in the `intensity` and `color` attributes respectively.

5.4.5 Additional validation rules

5.4.5.1 Introduction

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

5.4.5.2 If `intensity-value` is present, `intensity-range` must be present and vice versa.

5.4.5.3 The `intensity-value` must be within the `intensity-range`.

5.4.6 Example

EXAMPLE 1 This example shows the description of a light effect with the following semantics. The intensity is 50.0 lux (within a range of [10^{-5},32000] lux), i.e., approximately a family living room, with the color #FF0000. The light effect is activated at si:pts="0" and deactivated at si:pts="28".

```
<sedl:Effect xsi:type="sev:LightType" intensity-value="50.0" intensity-
range="0.00001 32000.0" activate="true" color="#FF0000" si:pts="0"/>
...
< sedl:Effect xsi:type="sev:LightType" activate="false" color="#FF0000"
si:pts="28"/>
```

EXAMPLE 2 This example shows the description of a light effect which uses the color scheme where the color is defined as a reference to the color classification scheme.

```
<sedl:DescriptionMetadata>
 <sedl:ClassificationSchemeAlias href="urn:mpeg:mpeg-v:01-SI-ColorCS-NS"
alias="COLOR"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:LightType" intensity-value="50.0" intensity-
range="0.00001 32000.0" duration="28" color=":COLOR:amber" si:pts="0"/>
```

EXAMPLE 3 This example shows the description of a light effect which disables the light effect on the left:middle:front only.

```
<sedl:DescriptionMetadata>
 <sedl:ClassificationSchemeAlias href="urn:mpeg:mpeg-v:01-SI-LocationCS-NS"
alias="WCS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:LightType" location=":WCS:left:middle:front"
activate="false" si:pts="40" fade="1"/>
```

5.5 Flash effect

5.5.1 Introduction

This Subclause specifies syntax and semantics of a flash effect.

5.5.2 Syntax

```
  <!-- ################################################ -->
  <!--  SEV Flash type                                  -->
  <!-- ################################################ -->
  <complexType name="FlashType">
    <complexContent>
      <extension base="sev:LightType">
        <attribute name="frequency" type="positiveInteger" use="optional"/>
      </extension>
    </complexContent>
  </complexType>
```

5.5.3 Binary representation syntax

FlashType {	*Number of bits*	*Mnemonic*
LightType		LightType
frequencyFlag	1	bslbf
if(frequencyFlag) {		
frequency	8	uimsbf
}		
}		

5.5.4 Semantics

Semantics of the `FlashType`:

Name	*Definition*
`FlashType`	Tool for describing a flash effect.
`frequencyFlag`	This field, which is only present in the binary representation, indicates the presence of the `frequency` attribute. If it is 1 then the `frequency` attribute is present, otherwise the `frequency` attribute is not present.
`frequency`	Describes the number of flickering in times per second. EXAMPLE The value 10 means it will flicker 10 times for each second.

5.5.5 Example

EXAMPLE This example shows the description of a flash effect with the following semantics. The intensity is 20.0 klx (within a range of [10^{-5},32000] lux), i.e., approximately full daylight (but not direct sunlight) with a duration of 5 seconds. The light flickers 10 times per second and the effect starts at si:pts="0".

```
<sedl:Effect xsi:type="sev:FlashType" intensity-value="20000.0" intensity-
range="0.00001 32000.0" duration="5" frequency="10" si:pts="0"/>
```

5.6 Temperature effect

5.6.1 Introduction

This Subclause specifies syntax and semantics of a temperature effect.

5.6.2 Syntax

```
<!-- ################################################ -->
<!--  SEV Temperature type                            -->
<!-- ################################################ -->
<complexType name="TemperatureType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <attribute name="intensity-value" type="sedl:intensityValueType"
```

```
            use="optional"/>
          <attribute name="intensity-range" type="sedl:intensityRangeType"
            use="optional"/>
        </extension>
      </complexContent>
    </complexType>
```

5.6.3 Binary representation syntax

TemperatureType {	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
intensityValueFlag	1	bslbf
intensityRangeFlag	1	bslbf
if(intensityValueFlag) {		
intensityValue	32	fsfb
}		
if(intensityRangeFlag) {		
intensityRange[0]	32	fsfb
intensityRange[1]	32	fsfb
}		
}		

5.6.4 Semantics

Semantics of the `TemperatureType`:

Name	*Definition*
`TemperatureType`	Tool for describing a temperature effect.
`intensityValueFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityValue attribute. If it is 1 then the intensity-value attribute is present, otherwise the intensity-value attribute is not present.
`intensityRangeFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityRange attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
`intensity-value`	Describes the intensity of the temperature effect in terms of heating/cooling in Celsius.
`intensity-range`	Describes the range of the intensity value. EXAMPLE [0.0, 100.0] on the Celsius scale.

5.6.5 Additional validation rules

5.6.5.1 Introduction

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

5.6.5.2 If `intensity-value` is present, `intensity-range` must be present and vice versa.

5.6.5.3 The `intensity-value` must be within the `intensity-range`.

5.6.6 Example

EXAMPLE This example shows the description of a temperature effect with the following semantics. The intensity is 10° (within a range of [-10.0,100.0]°), i.e., relatively cold, with a duration of 2 seconds and the effect starts at si:pts="0".

```
<sedl:Effect xsi:type="sev:TemperatureType" intensity-value="10.5" intensity-
range="-10.0 100.0" duration="2" si:pts="0"/>
```

5.7 Wind effect

5.7.1 Introduction

This Subclause specifies syntax and semantics of a wind effect.

5.7.2 Syntax

```
<!-- ################################################ -->
<!--  SEV Wind type                                   -->
<!-- ################################################ -->
<complexType name="WindType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <attribute name="intensity-value" type="sedl:intensityValueType"
         use="optional"/>
      <attribute name="intensity-range" type="sedl:intensityRangeType"
         use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

5.7.3 Binary representation syntax

WindType{	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
intensityValueFlag	1	bslbf
intensityRangeFlag	1	bslbf
if(intensityValueFlag) {		
intensityValue	32	fsfb
}		

if(intensityRangeFlag) {		
intensityRange[0]	32	fsfb
intensityRange[1]	32	fsfb
}		
}		

5.7.4 Semantics

Semantics of the `WindType`:

Name	*Definition*
`WindType`	Tool for describing a wind effect.
`intensityValueFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityValue attribute. If it is 1 then the intensity-value attribute is present, otherwise the intensity-value attribute is not present.
`intensityRangeFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityRange attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
`intensity-value`	Describes the intensity of the wind effect in terms of strength in Beaufort.
`intensity-range`	Describes the range of the intensity value. EXAMPLE [0.0, 12.0] on the Beaufort scale.

5.7.5 Additional validation rules

5.7.5.1 Introduction

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

5.7.5.2 If `intensity-value` is present, `intensity-range` must be present and vice versa.

5.7.5.3 The `intensity-value` must be within the `intensity-range`.

5.7.6 Example

EXAMPLE This example shows the description of a wind effect with the following semantics. The intensity is 3.0 according to the Beaufort scale (within a range of [0.0 12.0] Beaufort), i.e., approximately a gently breeze, with a fade-in time of 5 seconds and the effect is activated at si:pts="0".

```
<sedl:DescriptionMetadata>
 <sedl:ClassificationSchemeAlias alias="WCS" href="urn:mpeg:mpeg-v:01-SI-
LocationCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:WindType" fade="5" location=":WCS:*:*:front"
intensity-value="3.0" intensity-range="0.0 12.0" activate="true" si:pts="0"/>
```

5.8 Vibration effect

5.8.1 Introduction

This Subclause specifies syntax and semantics of a vibration effect.

5.8.2 Syntax

```
<!-- ################################################ -->
<!--  SEV Vibration type                              -->
<!-- ################################################ -->
<complexType name="VibrationType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <attribute name="intensity-value" type="sedl:intensityValueType"
         use="optional"/>
      <attribute name="intensity-range" type="sedl:intensityRangeType"
         use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

5.8.3 Binary representation syntax

VibrationType{	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
intensityValueFlag	1	bslbf
intensityRangeFlag	1	bslbf
if(intensityValueFlag) {		
intensityValue	32	fsfb
}		
if(intensityRangeFlag) {		
intensityRange[0]	32	fsfb
intensityRange[1]	32	fsfb
}		
}		

5.8.4 Semantics

Semantics of the `VibrationType`:

Name	*Definition*
`VibrationType`	Tool for describing a vibration effect.
`intensityValueFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityValue attribute. If it is 1 then the intensity-value attribute is present, otherwise the intensity-value attribute is not present.

Name	*Definition*
`intensityRangeFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityRange attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
`intensity-value`	Describes the intensity of the vibration effect in terms of thefrequency according to the Hz scale.
`intensity-range`	Describes the range of the intensity value. EXAMPLE [0.0, 50.0] on the Hz scale

5.8.5 Additional validation rules

5.8.5.1 Introduction

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

5.8.5.2 If `intensity-value` is present, `intensity-range` must be present and vice versa.

5.8.5.3 The `intensity-value` must be within the `intensity-range`.

5.8.6 Example

EXAMPLE This example shows the description of a vibration effect with the following semantics. The intensity is 4.1 according to the Hz scale (within a range of [0.0,50.0]), i.e., noticeable shaking items and rattling noises, with a duration of 7 seconds. The vibration has a fade-out time of 3 second and the effect starts at si:pts="0".

```
<sedl:Effect xsi:type="sev:VibrationType" intensity-value="4.1" intensity-
range="0.0 50.0" duration="7" fade="3" si:pts="0"/>
```

5.9 Spraying effect

5.9.1 Introduction

This Subclause specifies syntax and semantics of a spraying effect.

5.9.2 Syntax

```
<!-- ################################################ -->
<!--  Definition of Spraying type                     -->
<!-- ################################################ -->
<complexType name="SprayingType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <attribute name="intensity-value" type="sedl:intensityValueType"
         use="optional"/>
      <attribute name="intensity-range" type="sedl:intensityRangeType"
         use="optional"/>
```

```
        <attribute name="sprayingType" type="mpeg7:termReferenceType"/>
      </extension>
    </complexContent>
  </complexType>
```

5.9.3 Binary representation syntax

SprayingType {	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
intensityValueFlag	1	bslbf
intensityRangeFlag	1	bslbf
if(intensityValueFlag) {		
intensityValue	32	fsfb
}		
if(intensityRangeFlag) {		
intensityRange[0]	32	fsfb
intensityRange[1]	32	fsfb
}		
SprayingType	8	bslbf
}		

5.9.4 Semantics

Semantics of the `SprayingType`:

Name	*Definition*
`SprayingType`	Tool for describing a spraying effect.
`intensityValueFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityValue attribute. If it is 1 then the intensity-value attribute is present, otherwise the intensity-value attribute is not present.
`intensityRangeFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityRange attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
`intensity-value`	Describes the intensity of the spraying effect in terms in ml/h.
`intensity-range`	Describes the range of the intensity value. EXAMPLE [0.0, 10.0] ml/h.
`sprayingType`	Describes the type of the spraying effect as a reference to a classification scheme term. A CS that may be used for this purpose is the `SprayingTypeCS` defined in Annex A of ISO/IEC 23005-6.

5.9.5 Additional validation rules

5.9.5.1 Introduction

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

5.9.5.2 If `intensity-value` is present, `intensity-range` must be present and vice versa.

5.9.5.3 The `intensity-value` must be within the `intensity-range`.

5.9.6 Example

EXAMPLE This example shows the description of a water sprayer effect with the following semantics. The intensity is 0.7 ml/h (within a range of [0.0,10.0] ml/h) with a duration of 2 seconds and the effect starts at si:pts="0".

```
<sedl:Effect xsi:type="sev:SprayingType" intensity-value="0.7" intensity-
range="0.0 10.0" duration="2" si:pts="0"/>
```

5.10 Scent effect

5.10.1 Introduction

This Subclause specifies syntax and semantics of a scent effect.

5.10.2 Syntax

```
<!-- ################################################ -->
<!--  Definition of Scent type                        -->
<!-- ################################################ -->
<complexType name="ScentType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <attribute name="scent" type="mpeg7:termReferenceType"
        use="optional"/>
      <attribute name="intensity-value" type="sedl:intensityValueType"
         use="optional"/>
      <attribute name="intensity-range" type="sedl:intensityRangeType"
         use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

5.10.3 Binary representation syntax

ScentType{	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
intensityValueFlag	1	bslbf
intensityRangeFlag	1	bslbf
if(intensityValueFlag) {		
intensityValue	32	fsfb

}		
if(intensityRangeFlag) {		
intensityRange[0]	32	fsfb
intensityRange[1]	32	fsfb
}		
Scent	16	bslbf
}		

5.10.4 Semantics

Semantics of the `ScentType`:

Name	*Definition*
`ScentType`	Tool for describing a scent effect.
`intensityValueFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityValue attribute. If it is 1 then the intensity-value attribute is present, otherwise the intensity-value attribute is not present.
`intensityRangeFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityRange attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
`scent`	Describes the scent to use. A CS that may be used for this purpose is the `ScentCS` defined in Annex A of ISO/IEC 23005-6.
`intensity-value`	Describes the intensity of the scent effect in ml/h
`intensity-range`	Describes the range of the intensity value. EXAMPLE [0.0, 10.0] ml/h.

5.10.5 Additional validation rules

5.10.5.1 Introduction

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

5.10.5.2 If `intensity-value` is present, `intensity-range` must be present and vice versa.

5.10.5.3 The `intensity-value` must be within the `intensity-range`.

5.10.6 Example

EXAMPLE This example shows the description of a scent effect with the following semantics. The scent is lilac according to the classification scheme, the intensity is 0.1 ml/h (within a range of [0.0,10.0] ml/h) with a duration of 10 seconds and the effect starts at si:pts="0".

```
<sedl:DescriptionMetadata>
 <sedl:ClassificationSchemeAlias alias="SCENT" href="urn:mpeg:mpeg-v:01-SI-
ScentCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:ScentType" intensity-value="0.1" intensity-
range="0.0 10.0" duration="10" scent=":SCENT:lilac" si:pts="0"/>
```

5.11 Fog effect

5.11.1 Introduction

This Subclause specifies syntax and semantics of a fog effect.

5.11.2 Syntax

```
  <!-- ################################################ -->
  <!--  Definition of Fog type                          -->
  <!-- ################################################ -->
  <complexType name="FogType">
    <complexContent>
      <extension base="sedl:EffectBaseType">
        <attribute name="intensity-value" type="sedl:intensityValueType"
           use="optional"/>
        <attribute name="intensity-range" type="sedl:intensityRangeType"
           use="optional"/>
      </extension>
    </complexContent>
  </complexType>
```

5.11.3 Binary representation syntax

FogType{	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
intensityValueFlag	1	bslbf
intensityRangeFlag	1	bslbf
if(intensityValueFlag){		
intensityValue	32	fsfb
}		
if(intensityRangeFlag) {		
intensityRange[0]	32	fsfb
intensityRange[1]	32	fsfb
}		
}		

5.11.4 Semantics

Semantics of the `FogType`:

Name	*Definition*
FogType	Tool for describing a fog effect.
intensityValueFlag	This field, which is only present in the binary representation, indicates the presence of the intensityValue attribute. If it is 1 then the intensity-value attribute is present, otherwise the intensity-value attribute is not present.
intensityRangeFlag	This field, which is only present in the binary representation, indicates the presence of the intensityRange attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
intensity-value	Describes the intensity of the fog effect in ml/h.
intensity-range	Describes the range of the intensity value. EXAMPLE [0.0, 10.0] ml/h.

5.11.5 Additional validation rules

5.11.5.1 Introduction

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

5.11.5.2 If `intensity-value` is present, `intensity-range` must be present and vice versa

5.11.5.3 The `intensity-value` must be within the `intensity-range`.

5.11.6 Example

EXAMPLE This example shows the description of a fog effect with the following semantics. The intensity is 1.5 ml/h (within a range of [0.0,10.0] ml/h) with a duration of 20 seconds and the effect starts at si:pts="0".

```
<sedl:Effect xsi:type="sev:FogType" intensity-value="1.5" duration="20"
si:pts="0"/>
```

5.12 Color correction effect

5.12.1 Introduction

This Subclause specifies syntax and semantics of a color correction effect.

5.12.2 Syntax

```
<!-- ################################################ -->
<!--  Definition of Color Correction type             -->
<!-- ################################################ -->
<complexType name="ColorCorrectionType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <choice minOccurs="0">
```

```
          <element name="SpatioTemporalLocator"
type="mpeg7:SpatioTemporalLocatorType"/>
          <element name="SpatioTemporalMask"
type="mpeg7:SpatioTemporalMaskType"/>
        </choice>
        <attribute name="intensity-value" type="sedl:intensityValueType"
           use="optional"/>
        <attribute name="intensity-range" type="sedl:intensityRangeType"
           use="optional" fixed="0 1"/>
      </extension>
    </complexContent>
  </complexType>
```

5.12.3 Binary representation syntax

ColorCorrectionType{	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
intensityValueFlag	1	bslbf
intensityRangeFlag	1	bslbf
regionChoiceFlag	1	bslbf
If(regionChoiceFlag) {		
regionTypeChoice	1	bslbf
if(regionTypeChoice){		
SpatioTemporalLocator		SpatioTemporalLocatorType
}		
else{		
SpatioTemporalMask	See ISO 10646	UTF-8
}		
}		
if(intensityValueFlag) {		
Intensity-value	32	fsbf
}		
if(intensityRangeFlag) {		
Intensity-range	64	fsbf
}		
}		

5.12.4 Semantics

Semantics of the `ColorCorrectionType`:

Name	*Definition*
`ColorCorrectionType`	Tool for describing a color correction effect.
`intensityValueFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityValue attribute. If it is 1 then the intensity-value

Name	*Definition*
	attribute is present, otherwise the intensity-value attribute is not present.
`intensityRangeFlag`	This field, which is only present in the binary representation, indicates the presence of the intensityRange attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
`regionChoiceFlag`	This field, which is only present in the binary representation, indicates the presence of the choice. If it is 1 then the choice is present, otherwise the choice is not present.
`regionTypeChoice`	This field, which is only present in the binary representation, specifies the choice of the spatio-temporal region types. If it is 1 then the SpatioTemporalLocator is present, otherwise the SpatioTemporalMask is present.
`intensity-value`	Describes the intensity of the color correction effect in terms of “on” and “off” with respect to 1(on) and 0(off).
`intensity-range`	Describes the range of the intensity value, i.e., 1 (on) and 0 (off).
`SpatioTemporalLocator`	Describes the spatio-temporal localization of the moving region using `mpeg7:SpatioTemporalLocatorType` (optional), which indicates the regions in a video segment where the color correction effect is applied. The `mpeg7:SpatioTemporalLocatorType` is defined in ISO/IEC 15938-5.
`SpatioTemporalMask`	Describes a spatio-temporal mask that defines the spatio-temporal composition of the moving region (optional), which indicates the masks in a video segment where the color correction effect is applied. The `mpeg7:SpatioTemporalMaskType` is defined in ISO/IEC 15938-5.

5.12.5 Additional validation rules

5.12.5.1 Introduction

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

5.12.5.2 If `intensity-value` is present, `intensity-range` must be present and vice versa.

NOTE `intensity-range` has a `fixed` attribute and, thus, it must not be present explicitly as it is present implicitly.

5.12.5.3 The `intensity-value` must be within the `intensity-range`.

5.12.6 Example

EXAMPLE The following example shows a possible usage of the color correction effect applied to spatio-temporal regions as depicted in Figure 5.

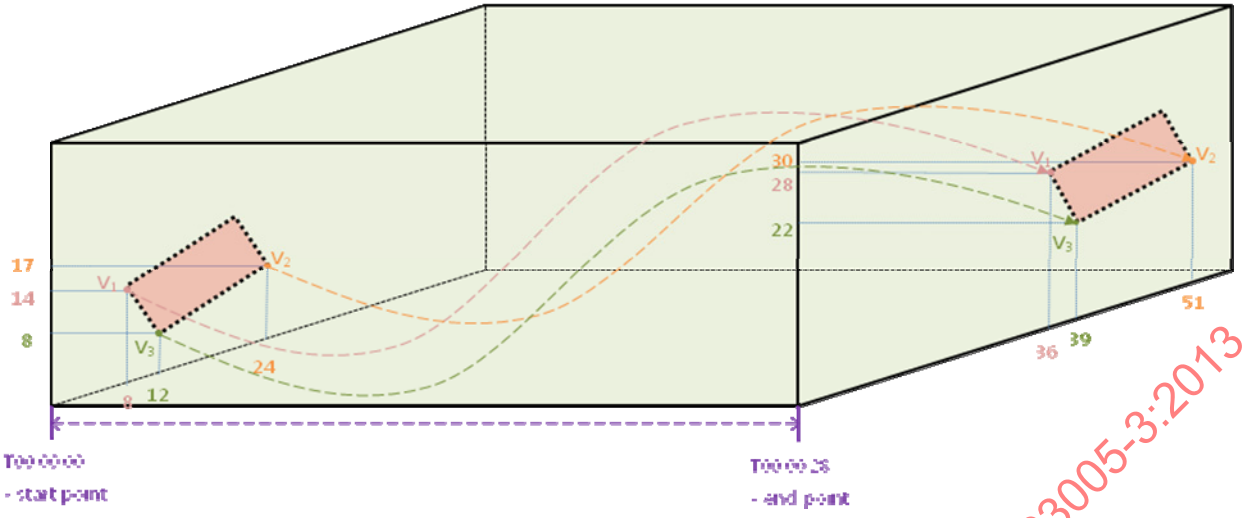


Figure 5 — Color correction effect applied to spatio-temporal regions

```
<sedl:Effect xsi:type="sev:ColorCorrectionType" intensity-value="1" intensity-
range="0 1" duration="28" si:pts="12000">
  <sev:SpatioTemporalLocator>
    <mpeg7:FigureTrajectory type="rectangle">
      <mpeg7:MediaTime>
        <mpeg7:MediaTimePoint>T00:00:00</mpeg7:MediaTimePoint>
        <mpeg7:MediaDuration>PT28S</mpeg7:MediaDuration>
      </mpeg7:MediaTime>
      <!-- Vertex 1 -->
      <mpeg7:Vertex>
        <mpeg7:KeyTimePoint>
          <mpeg7:MediaTimePoint>T00:00:00</mpeg7:MediaTimePoint>
          <mpeg7:MediaTimePoint>T00:00:28</mpeg7:MediaTimePoint>
        </mpeg7:KeyTimePoint>
        <!-- x coordinate -->
        <mpeg7:InterpolationFunctions>
          <mpeg7:KeyValue type="startPoint">8</mpeg7:KeyValue>
          <mpeg7:KeyValue type="secondOrder" param="1.0">36</mpeg7:KeyValue>
        </mpeg7:InterpolationFunctions>
        <!-- y coordinate -->
        <mpeg7:InterpolationFunctions>
          <mpeg7:KeyValue>14</mpeg7:KeyValue>
          <mpeg7:KeyValue>28</mpeg7:KeyValue>
        </mpeg7:InterpolationFunctions>
      </mpeg7:Vertex>
      <!-- Vertex 2 -->
      <mpeg7:Vertex>
        <mpeg7:KeyTimePoint>
          <mpeg7:MediaTimePoint>T00:00:00</mpeg7:MediaTimePoint>
          <mpeg7:MediaTimePoint>T00:00:28</mpeg7:MediaTimePoint>
        </mpeg7:KeyTimePoint>
        <!-- x coordinate -->
        <mpeg7:InterpolationFunctions>
          <mpeg7:KeyValue type="startPoint">24</mpeg7:KeyValue>
          <mpeg7:KeyValue type="secondOrder" param="1.0">51</mpeg7:KeyValue>
        </mpeg7:InterpolationFunctions>
        <!-- y coordinate -->
        <mpeg7:InterpolationFunctions>
          <mpeg7:KeyValue>17</mpeg7:KeyValue>
          <mpeg7:KeyValue>30</mpeg7:KeyValue>
        </mpeg7:InterpolationFunctions>
```

```
      </mpeg7:Vertex>
      <!-- Vertex 3 -->
      <mpeg7:Vertex>
        <mpeg7:KeyTimePoint>
          <mpeg7:MediaTimePoint>T00:00:00</mpeg7:MediaTimePoint>
          <mpeg7:MediaTimePoint>T00:00:28</mpeg7:MediaTimePoint>
        </mpeg7:KeyTimePoint>
        <!-- x coordinate -->
        <mpeg7:InterpolationFunctions>
          <mpeg7:KeyValue type="startPoint">12</mpeg7:KeyValue>
          <mpeg7:KeyValue type="secondOrder" param="1.0">39</mpeg7:KeyValue>
        </mpeg7:InterpolationFunctions>
        <!-- y coordinate -->
        <mpeg7:InterpolationFunctions>
          <mpeg7:KeyValue>8</mpeg7:KeyValue>
          <mpeg7:KeyValue>22</mpeg7:KeyValue>
        </mpeg7:InterpolationFunctions>
      </mpeg7:Vertex>
    </mpeg7:FigureTrajectory>
  </sev:SpatioTemporalLocator>
</sedl:Effect>
```

5.13 Rigid body motion effect

5.13.1 Introduction

This Subclause specifies syntax and semantics of a rigid body motion effect.

5.13.2 Syntax

```
  <!-- ################################################ -->
  <!--  Definition of Rigid Body Motion type            -->
  <!-- ################################################ -->
  <complexType name="RigidBodyMotionType">
    <complexContent>
      <extension base="sedl:EffectBaseType">
        <sequence>
          <element name="MoveToward" type="sev:MoveTowardType"
                   minOccurs="0"/>
          <element name="TrajectorySamples" type="sev:TrajectorySamples"
minOccurs="0" maxOccurs="unbounded"/>
          <element name="Incline" type="sev:InclineType" minOccurs="0"/>
          <element name="Shake" type="sev:ShakeType" minOccurs="0"/>
          <element name="Wave" type="sev:WaveType" minOccurs="0"/>
          <element name="Spin" type="sev:SpinType" minOccurs="0"/>
          <element name="Turn" type="sev:TurnType" minOccurs="0"/>
          <element name="Collide" type="sev:CollideType" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- ################################################ -->
  <!--  Definition of Move Toward type                  -->
  <!-- ################################################ -->
<complexType name="MoveTowardType">
<attribute name="speed" type="float" use="optional"/>
    <attribute name="acceleration" type="float" use="optional"/>
```

```
    <attribute name="directionV" type="sev:MoveTowardAngleType" use="optional"
default="0"/>
    <attribute name="directionH" type="sev:MoveTowardAngleType" use="optional"
default="0"/>
    <attribute name="distance-range" type="sedl:intensityRangeType"
use="optional"/>
    <attribute name="distance" type="float" use="optional"/>
  </complexType>

<!-- ################################################ -->
  <!--  Definition of Trajectory Samples type         -->
  <!-- ################################################ -->
<complexType name="TrajectorySamples">
```

```
    <complexContent>
      <extension base="mpeg7:FloatMatrixType">
        <attribute name="intensity-range" type="sedl:intensityRangeType"
use="optional"/>
      </extension>
```

```
  </complexContent>
</complexType>

  <!-- ################################################ -->
  <!--  Definition of Incline type                      -->
  <!-- ################################################ -->
<complexType name="InclineType">
<attribute name="pitchSpeed" type="float" use="optional"/>
    <attribute name="pitchAcceleration" type="float" use="optional"/>
    <attribute name="rollSpeed" type="float" use="optional"/>
    <attribute name="rollAcceleration" type="float" use="optional"/>
    <attribute name="yawSpeed" type="float" use="optional"/>
    <attribute name="yawAcceleration" type="float" use="optional"/>
<attribute name="pitch-range" type="sedl:intensityRangeType"
use="optional"/>
<attribute name="roll-range" type="sedl:intensityRangeType"
use="optional"/>
<attribute name="yaw-range" type="sedl:intensityRangeType"
use="optional"/>
    <attribute name="pitch" type="sev:InclineAngleType" use="optional"
default="0"/>
<attribute name="roll" type="sev:InclineAngleType" use="optional" default="0"/>
<attribute name="yaw" type="sev:InclineAngleType" use="optional" default="0"/>
  </complexType>

  <!-- ################################################ -->
  <!--  Definition of Shake type                        -->
  <!-- ################################################ -->
<complexType name="ShakeType">
<attribute name="direction" type="mpeg7:termReferenceType"
              use="optional"/>
    <attribute name="count" type="float" use="optional"/>
<attribute name="distance-range" type="sedl:intensityRangeType"
use="optional"/>
    <attribute name="distance" type="float" use="optional"/>
<attribute name="interval" type="positiveInteger" use="optional"/>
  </complexType>

  <!-- ################################################ -->
  <!--  Definition of Wave type                         -->
  <!-- ################################################ -->
```

```
<complexType name="WaveType">
<attribute name="direction" type="mpeg7:termReferenceType"
               use="optional"/>
    <attribute name="startDirection" type="mpeg7:termReferenceType"
               use="optional"/>
    <attribute name="count" type="float" use="optional"/>
<attribute name="distance-range" type="sedl:intensityRangeType"
use="optional"/>
    <attribute name="distance" type="float" use="optional"/>
<attribute name="interval" type="positiveInteger" use="optional"/>
  </complexType>

  <!-- ################################################ -->
  <!--  Definition of Spin type                         -->
  <!-- ################################################ -->
<complexType name="SpinType">
<attribute name="direction" type="mpeg7:termReferenceType"
               use="optional"/>
    <attribute name="count" type="float" use="optional"/>
<attribute name="interval" type="positiveInteger" use="optional"/>
  </complexType>

  <!-- ################################################ -->
  <!--  Definition of Turn type                         -->
  <!-- ################################################ -->
  <complexType name="TurnType">
<attribute name="direction" type="sev:TurnAngleType" use="optional"/>
    <attribute name="speed" type="float" use="optional"/>
  </complexType>

  <!-- ################################################ -->
  <!--  Definition of Collide type                      -->
  <!-- ################################################ -->
  <complexType name="CollideType">
<attribute name="directionH" type="sev:MoveTowardAngleType"
               use="optional" default="0"/>
    <attribute name="directionV" type="sev:MoveTowardAngleType"
               use="optional" default="0"/>
    <attribute name="speed" type="float" use="optional"/>
  </complexType>

  <!-- ################################################ -->
  <!--  Definition of Rigid Body Motion base type       -->
  <!-- ################################################ -->
  <simpleType name="TurnAngleType">
    <restriction base="integer">
      <minInclusive value="-180"/>
      <maxInclusive value="180"/>
    </restriction>
  </simpleType>
  <simpleType name="InclineAngleType">
    <restriction base="integer">
      <minInclusive value="-359"/>
      <maxInclusive value="359"/>
    </restriction>
  </simpleType>
  <simpleType name="MoveTowardAngleType">
    <restriction base="integer">
      <minInclusive value="0"/>
      <maxInclusive value="359"/>
    </restriction>
```

```
  </simpleType>
```

5.13.3 Binary representation syntax

RigidBodyMotionEffect {	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
MoveTowardFlag	1	bslbf
TrajectorySamplesFlag	1	bslbf
InclineFlag	1	bslbf
ShakeFlag	1	bslbf
WaveFlag	1	bslbf
SpinFlag	1	bslbf
TurnFlag	1	bslbf
CollideFlag	1	bslbf
If(MoveTowardFlag) {		
MoveToward		MoveTowardType
}		
If(TrajectorySamplesFlag) {		
TrajectorySamples		TrajectorySamplesType
}		
If(InclineFlag) {		
Incline		InclineType
}		
If(ShakeFlag){		
Shake		ShakeType
}		
If(WaveFlag){		
Wave		WaveType
}		
If(SpinFlag){		
Spin		SpinType
}		
If(TurnFlag){		
Turn		TurnType
}		
If(CollideFlag){		
Collide		CollideType
}		
}		

MoveTowardType{	*Number of bits*	*Mnemonic*
speedFlag	1	bslbf
accelerationFlag	1	bslbf
directionVFlag	1	bslbf
directionHFlag	1	bslbf
distanceFlag	1	bslbf
distanceRangeFlag	1	bslbf
If(speedFlag) {		
speed	32	fsfb
}		
If(accelerationFlag) {		
acceleration	32	fsfb
}		
If(directionVFlag) {		
directionV	32	fsfb
}		
If(directionHFlag) {		
directionH	32	fsfb
}		
If(distanceFlag) {		
distance	32	fsfb
}		
If(distanceRangeFlag) {		
distance-range		intensityRangeType
}		
}		

TrajectorySamplesType{	Number of bits	Mnemonic
intensityRangeFlag	1	bslbf
If(intensityRangeFlag) {		
Intensity-range		intensityRangeType
}		
SizeOfIntensityRow	16	uimsbf
SizeOfIntensityColumn	4	uimsbf
for(k=0;k<(SizeOfIntensityRow* SizeOfIntensityColumn);k++) {		
ArrayIntensity[k]	32	fsfb

}		
}		

InclineType {	*Number of bits*	*Mnemonic*
pitchSpeedFlag	1	bslbf
pitchAccelerationFlag	1	bslbf
rollSpeedFlag	1	bslbf
rollAccelerationFlag	1	bslbf
yawSpeedFlag	1	bslbf
yawAccelerationFlag	1	bslbf
pitchRangeFlag	1	bslbf
yawRangeFlag	1	bslbf
RollRangeFlag	1	bslbf
pitchFlag	1	bslbf
rollFlag	1	bslbf
yawFlag	1	bslbf
If(pitchSpeedFlag) {		
pitchSpeed	32	fsfb
}		
If(pitchAccelerationFlag) {		
pitchAcceleration	32	fsfb
}		
If(rollSpeedFlag) {		
rollSpeed	32	fsfb
}		
If(rollAccelerationFlag) {		
rollAcceleration	32	fsfb
}		
If(yawSpeedFlag) {		
yawSpeed	32	fsfb
}		
If(yawAccelerationFlag) {		
yawAcceleration	32	fsfb
}		
If(pitchRangeFlag) {		
pitch-range		intensityRangeType
}		
If(rollRangeFlag) {		

roll-range		intensityRangeType
}		
If(yawRangeFlag) {		
yaw-range		intensityRangeType
}		
If(pitchFlag) {		
pitch	10	bslbf
}		
If(rollFlag) {		
roll	10	bslbf
}		
If(yawFlag) {		
yaw	10	bslbf
}		
}		

ShakeType{	*Number of bits*	*Mnemonic*
directionFlag	1	bslbf
countFlag	1	bslbf
distanceRangeFlag	1	bslbf
distanceFlag	1	bslbf
intervalFlag	1	bslbf
If(directionFlag){		
direction	3	bslbf
}		
If(countFlag){		
count	32	fsfb
}		
If(distanceRangeFlag) {		
distance-range		intensityRangeType
}		
If(distanceFlag){		
distance	32	fsfb
}		
If(intervalFlag){		
interval	32	fsfb
}		
}		

WaveType{	*Number of bits*	*Mnemonic*
directionFlag	1	bslbf
startDirectionFlag	1	bslbf
countFlag	1	bslbf
distanceRangeFlag	1	bslbf
distanceFlag	1	bslbf
intervalFlag	1	bslbf
If(directionFlag){		
direction	2	bslbf
}		
If(startDirectionFlag){		
startDirection	2	bslbf
}		
If(countFlag){		
count	32	fsfb
}		
If(distanceRangeFlag) {		
distance-range		intensityRangeType
}		
If(distanceFlag){		
distance	32	fsfb
}		
If(intervalFlag){		
interval	32	fsfb
}		
}		

SpinType {	*Number of bits*	*Mnemonic*
directionFlag	1	bslbf
countFlag	1	bslbf
intervalFlag	1	bslbf
If(directionFlag){		
direction	3	bslbf
}		
If(countFlag){		
count	32	fsfb
}		
If(intervalFlag){		

interval	32	fsfb
}		
}		

TurnType {	*Number of bits*	*Mnemonic*
directionFlag	1	bslbf
speedFlag	1	bslbf
If(directionFlag){		
direction	9	simsbf
}		
If(speedFlag){		
speed	32	fsfb
}		
}		

CollideType{	*Number of bits*	*Mnemonic*
speedFlag	1	bslbf
directionH	9	uimsbf
directionV	9	uimsbf
If(speedFlag){		
speed	32	fsfb
}		
}		

5.13.4 Semantics

Semantics of the `RigidBodyMotionType`:

Name	*Definition*
`RigidBodyMotionType`	Tool for describing a rigid body motion effect.
`MoveTowardFlag`	This field, which is only present in the binary representation, indicates the presence of the `MoveToward` element. If it is 1 then the `MoveToward` element is present, otherwise the `MoveToward` element is not present.

MoveTorward	This pattern covers three dimensional movement of 6DoF, which means changing the location without rotation. The type is sev:MoveTorwardType.

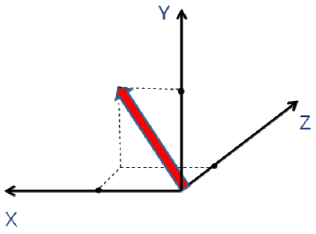


Figure 6 — MoveTorward pattern

TrajectorySamplesFlag	This field, which is only present in the binary representation, indicates the presence of the TrajectorySamples element. If it is 1 then the TrajectorySamples are present, otherwise the TrajectorySamples are not present.
TrajectorySamples	This pattern describes a set of position and orientation samples that the rigid body will follow. The trajectory sample data is comprised with *m* by 6 matrix, where 6 columns contain three positions (Px, Py, Pz in millimeters) and three orientations (Ox, Oy, Oz in degrees) and where m rows indicates the number of samples. The sampling rate shall be calculated by *m*/duration.

Figure 7 — TrajectorySamples pattern

InclineFlag	This field, which is only present in the binary representation, indicates the presence of the Incline element. If it is 1 then the Incline element is present, otherwise the Incline element is not present.
Incline	This pattern covers pitching, yawing, and rolling motion of 6 DoF, which means changing the rotation without changing the location. The type is sev:InclineType.

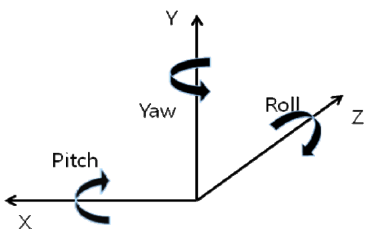
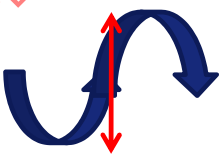
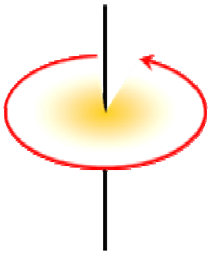


Figure 8 — Incline pattern

ShakeFlag	This field, which is only present in the binary representation, indicates the presence of the Shake element. If it is 1 then the Shake element is present, otherwise the Shake element is not present.
Shake	This pattern is a continuous motion moving from one side to opposite side repeatedly. This is an abstracted motion pattern which can be alternatively expressed by repetition of Move pattern. The type is sev:ShakeType  **Figure 9 — Shake pattern**
WaveFlag	This field, which is only present in the binary representation, indicates the presence of the Wave element. If it is 1 then the Wave element is present, otherwise the Wave element is not present.
Wave	This pattern is a continuous motion from side-up to side-down like the surface of water. This is an abstracted motion pattern which can be alternatively expressed by repetition of rolling or pitching of Incline pattern. The type is sev:WaveType.  **Figure 10 — Wave pattern**
SpinFlag	This field, which is only present in the binary representation, indicates the presence of the Spin element. If it is 1 then the Spin element is present, otherwise the Spin element is not present.
Spin	This pattern is a continuous turning based on a central point inside without change the place. This is an abstracted motion pattern which can be alternatively expressed by repetition of yawing of Incline pattern. The type is sev:SpinType  **Figure 11 — Spin pattern**
TurnFlag	This field, which is only present in the binary representation, indicates the presence of the Turn element. If it is 1 then the Turn element is present, otherwise the Turn element is not present.

Turn	This pattern is a motion of moving towards some direction. This is an abstracted motion pattern which can be alternatively expressed by repetition of Move and Incline pattern. The type is sev:TurnType.

Figure 12 — Turn pattern

CollideFlag	This field, which is only present in the binary representation, indicates the presence of the Collide element. If it is 1 then the Collide element is present, otherwise the Collide element is not present.
Collide	This pattern is a motion of moving object collides against something. This is an abstracted motion pattern which can be alternatively expressed by repetition of Move and Incline pattern. The type is sev:CollideType.

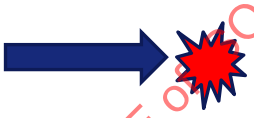


Figure 13 — Collide pattern

Semantics of the MoveTorwardType:

Name	*Definition*
speedFlag	This field, which is only present in the binary representation, indicates the presence of the speed attribute. If it is 1 then the speed attribute is present, otherwise the speed attribute is not present
Speed	Describes the moving speed in terms of centimeter per second.
accelerationFlag	This field, which is only present in the binary representation, indicates the presence of the acceleration attribute. If it is 1 then the acceleration attribute is present, otherwise the acceleration attribute is not present
Acceleration	Describes the acceleration in terms of centimeter per square second.
directionHFlag	This field, which is only present in the binary representation, indicates the presence of the directionH attribute. If it is 1 then the directionH attribute is present, otherwise the directionH attribute is not present

directionH	Describes the horizontal direction of moving in terms of angle. The type is sev:MoveTowardAngleType. The angle starts from the front-center of the rigid body and increases with CCW (Counter Clock Wise) .

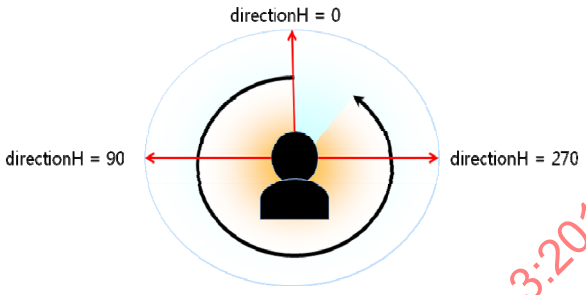


Figure 14 — Horizontal direction model for MoveToward pattern

directionVFlag	This field, which is only present in the binary representation, indicates the presence of the directionV attribute. If it is 1 then the directionV attribute is present, otherwise the directionV attribute is not present
directionV	Describes the vertical direction of moving in terms of angle. The type is sev:MoveTowardAngleType. The angle starts from the front-center of rigid body and increases with CCW.

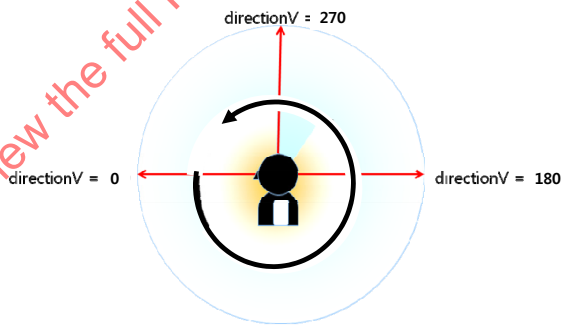


Figure 15 — Vertical direction model for MoveToward pattern

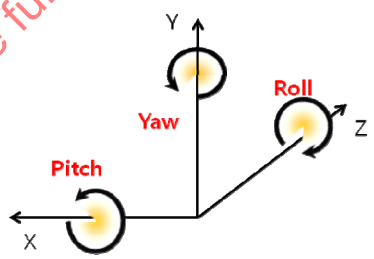
distanceFlag	This field, which is only present in the binary representation, indicates the presence of the distance attribute. If it is 1 then the distance attribute is present, otherwise the distance attribute is not present.
distance	Describes the distance between the origin and destination in terms of centimeter.
distanceRangeFlag	This field, which is only present in the binary representation, indicates the presence of the distance-range attribute. If it is 1 then the distance-range attribute is present, otherwise the distance-range attribute is not present.
distance-range	Describes the range of the distance value. EXAMPLE [-100, 100].

Semantics of the TrajectorySamplesType:

intensityRangeFlag	This field, which is only present in the binary representation, indicates the presence of the intensity-range attribute. If it is 1 then the intensity-range attribute is present, otherwise the intensity-range attribute is not present.
intensity-range	Describes the range of the intensity value. EXAMPLE [-100, 100].
SizeOfforceRow	Describes a row size of intensity
SizeOfforceColumn	Describes a column size of instensity (Usually 6)
ArrayIntensity	Describes intensities in terms of physical quantities for all elements of *m* by *n* matrix of the trajectorySample actuators.

Semantics of the InclineType:

Name	*Definition*
pitchSpeedFlag	This field, which is only present in the binary representation, indicates the presence of the pitchSpeed attribute. If it is 1 then the pitchSpeed attribute is present, otherwise the pitchSpeed attribute is not present.
PitchSpeed	Describes the rotation speed based on X-axis in terms of degree per second.
pitchAccelerationFlag	This field, which is only present in the binary representation, indicates the presence of the pitchAcceleration attribute. If it is 1 then the pitchAcceleration attribute is present, otherwise the pitchAcceleration attribute is not present.
PitchAcceleration	Describes the acceleration based on X-axis in terms of degree per square second.
rollSpeedFlag	This field, which is only present in the binary representation, indicates the presence of the rollSpeed attribute. If it is 1 then the rollSpeed attribute is present, otherwise the rollSpeed attribute is not present.
RollSpeed	Describes the rotation speed based on Z-axis in terms of degree per second.
rollAccelerationFlag	This field, which is only present in the binary representation, indicates the presence of the rollAcceleration attribute. If it is 1 then the rollAcceleration attribute is present, otherwise the rollAcceleration attribute is not present.
RollAcceleration	Describes the acceleration based on Z-axis in terms of degree per square second.
yawSpeedFlag	This field, which is only present in the binary representation, indicates the presence of the yawSpeed attribute. If it is 1 then the yawSpeed attribute is present, otherwise the yawSpeed attribute is not present.

YawSpeed	Describes the rotation speed based on Y-axis in terms of degree per second.
yawAccelerationFlag	This field, which is only present in the binary representation, indicates the presence of the yawAcceleration attribute. If it is 1 then the yawAcceleration attribute is present, otherwise the yawAcceleration attribute is not present.
YawAcceleration	Describes the acceleration based on Y-axis in terms of degree per square second.
pitchRangeFlag	This field, which is only present in the binary representation, indicates the presence of the pitch-range attribute. If it is 1 then the pitch-range attribute is present, otherwise the pitch-range attribute is not present.
pitch-range	Describes the range of the pitch value. EXAMPLE [-100, 100].
pitchFlag	This field, which is only present in the binary representation, indicates the presence of the pitchSpeed attribute. If it is 1 then the pitchSpeed attribute is present, otherwise the pitchSpeed attribute is not present.
pitch	Describes the rotation based on X-axis in terms of angle. Positive value means the rotation angle in the direction of pitch arrow.  **Figure 16 — Direction model for Incline pattern** NOTE The pitch angle is increased with counter-clock wise.
rollRangeFlag	This field, which is only present in the binary representation, indicates the presence of the roll-range attribute. If it is 1 then the roll-range attribute is present, otherwise the roll-range attribute is not present.
roll-range	Describes the range of the roll value. EXAMPLE [-100, 100].
rollFlag	This field, which is only present in the binary representation, indicates the presence of the roll attribute. If it is 1 then the roll attribute is present, otherwise the roll attribute is not present.
roll	Describes the rotation based on Z-axis in terms of angle. Positive value means the rotation angle in the direction of roll arrow as depicted in the Figure 16. NOTE The roll angle is increased with counter-clock wise.

yawRangeFlag	This field, which is only present in the binary representation, indicates the presence of the yaw-range attribute. If it is 1 then the yaw-range attribute is present, otherwise the yaw-range attribute is not present.
yaw-range	Describes the range of the yaw value. EXAMPLE [-100, 100].
yawFlag	This field, which is only present in the binary representation, indicates the presence of the yaw attribute. If it is 1 then the yaw attribute is present, otherwise the yaw attribute is not present.
yaw	Describes the rotation based on Y-axis in terms of angle. Positive value means the rotation angle in the direction of yaw arrow as depicted in the Figure 16. NOTE The yaw angle is increased with counter-clock wise.

Semantics of the ShakeType:

Name	*Definition*
directionFlag	This field, which is only present in the binary representation, indicates the presence of the direction attribute. If it is 1 then the direction attribute is present, otherwise the direction attribute is not present.
direction	Describes the direction of the shake motion. A CS that may be used for this purpose is the ShakeDirectionCS defined in Annex A of ISO/IEC 23005-6.
countFlag	This field, which is only present in the binary representation, indicates the presence of the count attribute. If it is 1 then the count attribute is present, otherwise the count attribute is not present.
count	Describes the times to shake during the duration time.

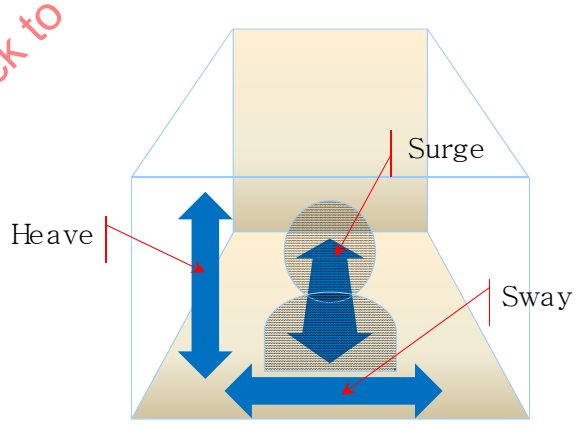
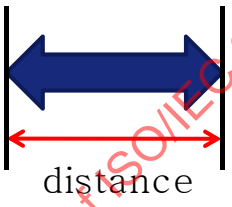
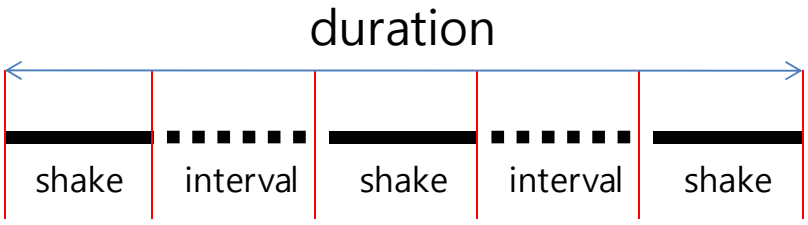


Figure 17 — Direction model for Shake pattern

distanceRangeFlag	This field, which is only present in the binary representation, indicates the presence of the distance-range attribute. If it is 1 then the distance-range attribute is present, otherwise the distance-range attribute is not present.
distance-range	Describes the range of the distance value. EXAMPLE [-100, 100].
distanceFlag	This field, which is only present in the binary representation, indicates the presence of the distance attribute. If it is 1 then the distance attribute is present, otherwise the distance attribute is not present.
distance	Describes the distance between the two ends of the shaking motion in terms of centimeter.  **Figure 18 — Distance for shaking motion**
intervalFlag	This field, which is only present in the binary representation, indicates the presence of the interval attribute. If it is 1 then the interval attribute is present, otherwise the interval attribute is not present.
interval	Describes a break time in the intervals of shake motions according to the time scheme used. The time scheme used shall be identified by means of the si:absTimeScheme and si:timeScale attributes respectively. NOTE The interval is illustrated in Figure 19.  **Figure 19 — Illustration of interval**

Semantics of the WaveType:

Name	*Definition*
directionFlag	This field, which is only present in the binary representation, indicates the presence of the direction attribute. If it is 1 then the direction attribute is present, otherwise the direction attribute is not present.

direction	Describes the direction of the wave motion. A CS that may be used for this purpose is the WaveDirectionCS defined in Annex A of ISO/IEC 23005-6.

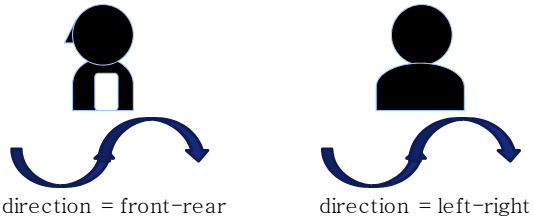


Figure 20 — Direction for waving motion

startDirectionFlag	This field, which is only present in the binary representation, indicates the presence of the startDirection attribute. If it is 1 then the startDirection attribute is present, otherwise the startDirection attribute is not present.
startDirection	Describes whether it starts towards up direction or down direction. A CS that may be used for this purpose is the WaveStartDirectionCS defined in Annex A of ISO/IEC 23005-6.

Figure 21 — Start direction for waving motion

countFlag	This field, which is only present in the binary representation, indicates the presence of the count attribute. If it is 1 then the count attribute is present, otherwise the count attribute is not present.
count	Describes the times to wave during the duration time.
distanceRangeFlag	This field, which is only present in the binary representation, indicates the presence of the distance-range attribute. If it is 1 then the distance-range attribute is present, otherwise the distance-range attribute is not present.
distance-range	Describes the range of the distance value. EXAMPLE [-100, 100].
distanceFlag	This field, which is only present in the binary representation, indicates the presence of the distance attribute. If it is 1 then the distance attribute is present, otherwise the distance attribute is not present.

distance	Describes the distance between the top and the bottom of the wave motion in centimeter.

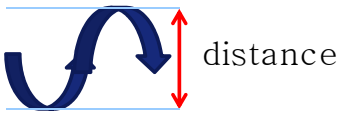


Figure 22 — Distance for waving motion

intervalFlag	This field, which is only present in the binary representation, indicates the presence of the interval attribute. If it is 1 then the interval attribute is present, otherwise the interval attribute is not present.
interval	Describes a break time in the intervals of wave motions according to the time scheme used. The time scheme used shall be identified by means of the si:absTimeScheme and si:timeScale attributes respectively.

Semantics of the TurnType:

Name	*Definition*
directionFlag	This field, which is only present in the binary representation, indicates the presence of the direction attribute. If it is 1 then the direction attribute is present, otherwise the direction attribute is not present.
direction	Describes the turning direction in terms of angle. The type is sev:TurnAngleType.

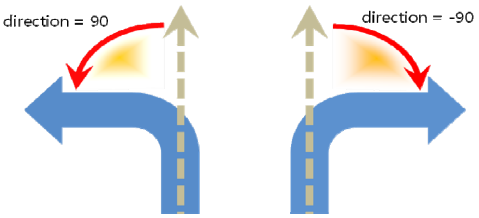


Figure 23 — Direction for turn pattern

speedFlag	This field, which is only present in the binary representation, indicates the presence of the speed attribute. If it is 1 then the speed attribute is present, otherwise the speed attribute is not present.
speed	Describes the turning speed in degree per second.

Semantics of the SpinType:

Name	*Definition*

directionFlag	This field, which is only present in the binary representation, indicates the presence of the direction attribute. If it is 1 then the direction attribute is present, otherwise the direction attribute is not present.
direction	Describes the direction of the spinning based on the 3 axes. A CS that may be used for this purpose is the SpinDirectionCS defined in Annex A of ISO/IEC 23005-6. NOTE 1 Forward-spin based on x axis (which is "xf" in the classification scheme) indicates the spinning direction by the pitch arrow depicted in the Figure 16. Otherwise, backward-spin based on x axis (which is "xb" in the classification scheme) indicates the opposite spinning direction of "xf".
countFlag	This field, which is only present in the binary representation, indicates the presence of the count attribute. If it is 1 then the count attribute is present, otherwise the count attribute is not present.
count	Describes the times to spin during the duration time.
intervalFlag	This field, which is only present in the binary representation, indicates the presence of the interval attribute. If it is 1 then the interval attribute is present, otherwise the interval attribute is not present.
interval	Describes a break time in the intervals of spin motions according to the time scheme used. The time scheme used shall be identified by means of the si:absTimeScheme and si:timeScale attributes respectively.

Semantics of the CollideType:

Name	*Definition*
directionH	Describes the horizontal direction of receiving impact in terms of angle. The type is sev:MoveTowardAngleType. The angle starts from the front-center of the rigid body and increases with CCW.

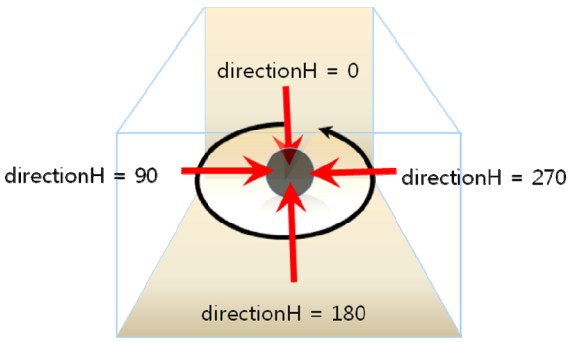


Figure 24 — Horizontal direction model for collide pattern

directionV	Describes the vertical direction of receiving impact in terms of angle. The type is sev:TowardAngleType. The angle starts from the front-center of rigid body and increases with CCW.

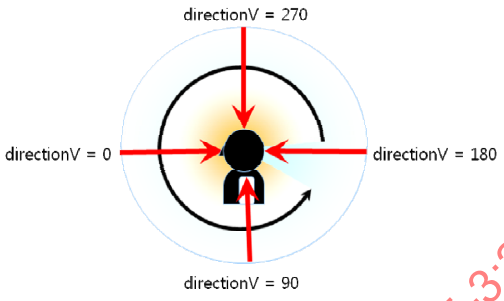


Figure 25 — Vertical direction model for collide pattern

speedFlag	This field, which is only present in the binary representation, indicates the presence of the speed attribute. If it is 1 then the speed attribute is present, otherwise the speed attribute is not present.
speed	Describes the speed of colliding object in terms of centimeter per second.

5.13.5 Examples

EXAMPLE 1 This example shows the description of a movetoward effect with the following semantics. The distance to the target position is 6 cm, and the direction is 30 degrees in the vertical angle and 50 degrees in the horizontal angle. The moving speed is 3 cm per second and the duration is 2 seconds. The rigid body starts to move towards target position at si:pts="0" and stays at the target position until the duration time..

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="2" si:pts="0">
  <sev:MoveToward directionV="30" directionH="50" distance="6"
     speed="3" />
</sedl:Effect>
```

It is possible to describe acceleration instead of speed.

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="2" si:pts="0">
  <sev:MoveToward directionV="30" directionH="50" distance="6"
     acceleration="3" />
</sedl:Effect>
```

It is also possible to describe without speed or acceleration. In this case the semantics is that the rigid body starts to movetoward at si:pts="0" and should reach at the target position within the duration time, which implies that the speed is decided by the duration and the distance. According to the below example, the implied speed is 3 cm/s.

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="2" si:pts="0">
  <sev:MoveToward directionV="30" directionH="50" distance="6"/>
</sedl:Effect>
```

EXAMPLE 2 This example shows the description of an incline effect with the following semantics. The target pitch angle is 30 degrees. The rotation speed is 10 degrees per second and the duration is 5 seconds. The rigid body starts to rotate towards target angle at si:pts="0" and stays at the target angle until the duration time.

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="5" si:pts="0">
  <sev:Incline pitch="30" pitchSpeed="3.0"/>
</sedl:Effect>
```

It is possible to describe acceleration instead of speed.

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="5" si:pts="0">
  <sev:Incline pitch="30" pitchAcceleration="3.0"/>
</sedl:Effect>
```

It is also possible to describe without speed or acceleration. In this case the semantics is that the rigid body starts to incline at si:pts="0" and should finish this motion during the duration time, which implies that the speed is decided by the duration and the angle.

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="5" si:pts="0">
  <sev:Incline pitch="30"/>
</sedl:Effect>
```

EXAMPLE 3 This example shows the description of a shake effect with the following semantics. The moving direction is sway which means moving left and right repeatedly. The end-to-end distance is 10 cm and the duration is 5 seconds. The rigid body starts to shake at si:pts="0" and shake 5 times until the duration time.

```
<sedl:DescriptionMetadata>
  <sedl:ClassificationSchemeAlias alias="SHAKE"
     href="urn:mpeg:mpeg-v:01-SI-ShakeDirectionCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="5" si:pts="0">
  <sev:Shake direction=":SHAKE:sway" distance="10" count="5"/>
</sedl:Effect>
```

EXAMPLE 4 This example shows the description of a wave effect with the following semantics. The wave direction is left-right and the distance is 10 cm, and the duration is 5 seconds. The rigid body starts to wave with the left side up at si:pts="0" and wave 5 times until the duration time.

```
<sedl:DescriptionMetadata>
  <sedl:ClassificationSchemeAlias alias="WAVE"
     href="urn:mpeg:mpeg-v:01-SI-WaveDirectionCS-NS"/>
<sedl:ClassificationSchemeAlias alias="WAVESTR"
     href="urn:mpeg:mpeg-v:01-SI-WaveStartDirectionCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="5" si:pts="0">
  <sev:Wave direction=":WAVE:left-right"
        startDirection=":WAVESTR:up" count="5" distance="10"/>
</sedl:Effect>
```

EXAMPLE 5 This example shows the description of a turn effect with the following semantics. The turn direction is 30 degree and the turning speed is 10 degree per seconds. The rigid body starts to turn (including move towards) in the right direction with the speed at si:pts="0" and stays during the duration time.

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="5" si:pts="0">
  <sev:Turn direction="30" speed="10"/>
</sedl:Effect>
```

It is also possible to describe without speed. In this case the semantics is that the rigid body starts to turn in the right direction at si:pts="0" and should finish this motion during the duration time.

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="5" si:pts="0">
  <sev:Turn direction="90"/>
</sedl:Effect>
```

EXAMPLE 6 This example shows the description of a spin effect with the following semantics. The spinning direction is “xf” which means forward spin based on x-axis. The spinning count is 5 times and the duration is 6 seconds which means the rigid body starts to spin at si:pts="0" and repeats 5 times for 6 seconds.

```
<sedl:DescriptionMetadata>
  <sedl:ClassificationSchemeAlias alias="SPIN"
     href="urn:mpeg:mpeg-v:01-SI-SpinDirectionCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:RigidBodyMotionType" duration="6" si:pts="0">
  <sev:Spin direction=":SPIN:xf" count="5"/>
</sedl:Effect>
```

EXAMPLE 7 This example shows the description of a collide effect with the following semantics. The rigid body receives the colliding impact from the horizontal angle 90 degrees and vertical angle 30 degrees with the speed 220 cm per second.

```
<sedl:Effect xsi:type="sev:RigidBodyMotionType" si:pts="0">
  <sev:Collide directionH="90" directionV="30" speed="220"/>
</sedl:Effect>
```

5.14 Passive kinesthetic motion effect

5.14.1 Introduction

This Subclause specifies syntax and semantics of a passive kinesthetic motion effect.

5.14.2 Syntax

```
<!-- ################################################ -->
<!--  SEV Passive Kinesthetic Motion type             -->
<!-- ################################################ -->
<complexType name="PassiveKinestheticMotionType">
  <complexContent>
    <extension base="sev:RigidBodyMotionType">
      <attribute name="updateRate" type="positiveInteger" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

5.14.3 Binary representation syntax

PassiveKinestheticMotioin {	*Number of bits*	*Mnemonic*
RigidBodyMotionType		RigidBodyMotionType
updateRate	16	uimsbf
}		

5.14.4 Semantics

Semantics of the `PassiveKinestheticMotionType`:

`Name`	Definition
`PassiveKinestheticMotionType`	Tool for describing a passive kinesthetic motion effect. This type defines a passive kinesthetic motion mode. In this mode, a user holds the kinesthetic device softly and the kinesthetic device guides the user's hand according to the recorded motion trajectories that are specified by three positions and three orientations.
`TrajectorySamples`	Tool for describing a passive kinesthetic interaction. The passive kinesthetic motion data is comprised with m by 6 matrix, where 6 columns contain three positions (Px, Py, Pz in millimeters) and three orientations (Ox, Oy, Oz in degrees) and where m rows contain timescale data points. These six data are updated with the same *m* timeScale.
`updateRate`	Describes a number of data update times per second. EXAMPLE The value 20 means that the kinesthetic device will move to 20 different positions and orientations for each second.

5.14.5 Additional validation rules

5.14.5.1 Introduction

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

5.14.5.2 A number of `TrajectorySamples` must be in multiples of 6 since 6 samples include three positions and three orientations at each period Therefore number of rows is fixed to 6.

5.14.5.3 A number of `TrajectorySamples` at an update period must not exceed the `duration` therefore `updateRate` attribute should be set.

5.14.6 Examples

EXAMPLE This example shows the description of a passive kinesthetic motion effect with the following semantics. The kinesthetic device moves to new positions and orientations according to the timeSamples. If 3 position and orientation data are given and updateRate is 3, then the kinesthetic device guides user's hand to three different positions and orientations for 1 second since 3 samples for position and orientation data are provided. Note that the passive kinesthetic motion effect is activated at si:pts="5".

```
<sedl:Effect xsi:type="sev:PassiveKinestheticMotionType" updateRate="3"
si:pts="5">
  <sev:TrajectorySamples mpeg7:dim="3 6">
     15 0  20 10 30 10      20 11 50 30 10  0
      0 10 10 10  0 15
  </sev:TrajectorySamples>
</sedl:Effect>
```

5.15 Passive kinesthetic force effect

5.15.1 Introduction

This Subclause specifies syntax and semantics of a passive kinesthetic force effect.

5.15.2 Syntax

```
<!-- ################################################ -->
<!--  SEV Passive Kinesthetic Force type              -->
<!-- ################################################ -->
<complexType name="PassiveKinestheticForceType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <sequence>
        <element name="PassiveKinestheticForce"
                 type="mpeg7:FloatMatrixType"/>
      </sequence>
      <attribute name="updateRate" type="positiveInteger" use="required"/>
    </extension>
  </complexContent>
</complexType>
```

5.15.3 Binary representation syntax

PassiveKinestheticForce{	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
SizeOfforceRow	16	uimsbf
SizeOfforceColumn	4	uimsbf
for(k=0;k<(SizeOfforceRow* SizeOfforceColumn);k++) {		
force[k]	32	fsfb
}		
updateRate	16	uimsbf
}		

5.15.4 Semantics

Semantics of the `PassiveKinestheticMotionType`:

Name	Definition

PassiveKinestheticForceType	Tool for describing a passive kinesthetic force/torque effect. This type defines a passive kinesthetic force/torque mode. In this mode, a user holds the kinesthetic device softly and the kinesthetic device guides the user's hand according to the recorded force/toque histories.
PassiveKinestheticForce	Describes a passive kinesthetic force/torque sensation. The passive kinesthetic force/torque data are comprised with m by 6 matrix, where 6 columns contain three forces (Fx, Fy, Fz in Newton) and three torques (Tx, Ty, Tz in Newton-millimeter) for force/torque trajectories. These six data are updated with the updateRate.
SizeOfforceRow	Describes a row size of force
SizeOfforceColumn	Describes a column size of force(Usually 6)
force	Describes m by 6 matrix, where 6 columns contain three forces (Fx, Fy, Fz in Newton) and three torques (Tx, Ty, Tz in Newton-millimeter) for force/torque trajectories. 'm' represents the number of position samples.
updateRate	Describes a number of data update times per second.

5.15.5 Additional validation rules

5.15.5.1 Introduction

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

5.15.5.2 A data number of `PassiveKinestheticForce` must be in multiples of 6 since 6 samples include three forces and three torques at each period. Therefore number of columns is fixed to 6

5.15.5.3 `PassiveKinestheticForce` effect element must have `updateRate` attribute in parent element, but `duration` attribute must not be described.

5.15.5.4 `PassiveKinestheticForce` effect element must not be described with `ActiveKinesthetic` effect at the same time.

5.15.6 Examples

EXAMPLE This example shows the description of a passive kinesthetic force effect with the following semantics. The kinesthetic device produces new forces and torques according to the update rate. If 3 forces and torques data are given and updateRate is 3, then the kinesthetic device produces forces and torques for 1 second since 3 samples for position and orientation data are provided. Note that the passive kinesthetic force effect is activated at si:pts="2".

```
<sedl:Effect xsi:type="sev:PassiveKinestheticForceType" updateRate="3"
si:pts="2">
  <sev:PassiveKinestheticForce mpeg7:dim="3 6">
     3  0 0 0  3 0
     0  0 3 2  0 0
     0  0 3 0 10 5
  </sev:Passivekinestheticforce>
</sedl:Effect>
```

5.16 Active kinesthetic effect

5.16.1 Introduction

This Subclause specifies syntax and semantics of an active kinesthetic effect.

5.16.2 Syntax

```
<!-- ################################################ -->
<!--  SEV Active  Kinesthetic type                    -->
<!-- ################################################ -->
<complexType name="ActiveKinestheticType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <sequence>
        <element name="ActiveKinesthetic"
                 type="sev:ActiveKinestheticForceType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="ActiveKinestheticForceType">
  <attribute name="fx" type="float" use="optional"/>
  <attribute name="fy" type="float" use="optional"/>
  <attribute name="fz" type="float" use="optional"/>
  <attribute name="tx" type="float" use="optional"/>
  <attribute name="ty" type="float" use="optional"/>
  <attribute name="tz" type="float" use="optional"/>
</complexType>
```

5.16.3 Binary representation syntax

ActiveKinesthetic {	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
txFlag	1	bslbf
tyFlag	1	bslbf
tzFlag	1	bslbf
fxFlag	1	bslbf
fyFlag	1	bslbf
fzFlag	1	bslbf
if(txFlag) {		
tx	32	fsfb
}		
if(tyFlag) {		
ty	32	fsfb
}		
if(tzFlag) {		
tz	32	fsfb

}		
if(fxFlag) {		
fx	32	fsfb
}		
If(fyFlag) {		
fy	32	fsfb
}		
If(fzFlag) {		
fz	32	fsfb
}		
}		

5.16.4 Semantics

Semantics of the `ActiveKinestheticType`:

Name	Definition
`ActiveKinestheticType`	Tool for describing an active kinesthetic effect. This type defines an active kinesthetic interaction mode. In this mode, when a user touches an object by his/her will, then the computed contact forces and torques are provided.
`ActiveKinesthetic`	Describes an active kinesthetic interaction.
`ActiveKinestheticForceType`	Tool for describing three forces (fx, fy, fz) and torques (tx, ty, tz) for each axis in an active kinesthetic mode. Force is represented in the unit of N (Newton) and torque is represented in the unit of Nmm (Newton-millimeter).
`txFlag`	This field, which is only present in the binary representation, indicates the presence of the `tx` attribute. If it is 1 then the `tx` attribute is present, otherwise the `tx` attribute is not present.
`tyFlag`	This field, which is only present in the binary representation, indicates the presence of the `ty` attribute. If it is 1 then the `ty` attribute is present, otherwise the `ty` attribute is not present.
`tzFlag`	This field, which is only present in the binary representation, indicates the presence of the `tz` attribute. If it is 1 then the `tz` attribute is present, otherwise the `tz` attribute is not present.
`fxFlag`	This field, which is only present in the binary representation, indicates the presence of the `fx` attribute. If it is 1 then the `fx` attribute is present, otherwise the `fx` attribute is not present.
`fyFlag`	This field, which is only present in the binary representation, indicates the presence of the `fy` attribute. If it is 1 then the `fy` attribute is present, otherwise the `fy` attribute is not present.

fzFlag	This field, which is only present in the binary representation, indicates the presence of the fz attribute. If it is 1 then the fz attribute is present, otherwise the fz attribute is not present.
tx	Torque for x-axis in an active kinesthetic mode. Torque is represented in the unit of Nmm(Newton-millimeter).
ty	Torque for y-axis in an active kinesthetic mode. Torque is represented in the unit of Nmm(Newton-millimeter).
tz	Torque for z-axis in an active kinesthetic mode. Torque is represented in the unit of Nmm(Newton-millimeter).
fx	Force for x-axis in an active kinesthetic mode. Force is represented in the unit of N(Newton).
fy	Force for y-axis in an active kinesthetic mode. Force is represented in the unit of N(Newton).
fz	Force for z-axis in an active kinesthetic mode. Force is represented in the unit of N(Newton).

5.16.5 Examples

EXAMPLE This example shows the description of an active kinesthetic effect with the following semantics. Forces and torques are given with the computed values by a user's input, where unit of force is N(newton) and unit of torque is Nmm.

```
<sedl:Effect xsi:type="sev:ActiveKinestheticType">
  <sev:ActiveKinesthetic  fx="1.0" fy="2.0" fz="1.5" tx="200" ty="170"
tz="300"/>
</sedl:Effect>
```

5.17 Tactile effect

5.17.1 Introduction

This Subclause specifies syntax and semantics of a tactile effect.

5.17.2 Syntax

```
<!-- ################################################ -->
<!--  SEV Tactile type                                -->
<!-- ################################################ -->
<complexType name="TactileType">
  <complexContent>
    <extension base="sedl:EffectBaseType">
      <sequence>
        <choice>
          <element name="ArrayIntensity" type="mpeg7:FloatMatrixType"/>
          <element name="TactileResource" type="sev:TactileResourceType"/>
        </choice>
      </sequence>
```

```
        <attribute name="tactileEffect" type="mpeg7:termReferenceType"
                   use="required"/>
        <attribute name="updateRate" type="positiveInteger" use="optional"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- ################################################ -->
  <!--  SEV TactileResource type                        -->
  <!-- ################################################ -->
  <complexType name="TactileResourceType">
    <attribute name="type" type="string" use="required"/>
    <attribute name="ref" type="anyURI" use="required"/>
  </complexType>
```

5.17.3 Binary representation syntax

Tactile {	*Number of bits*	*Mnemonic*
EffectBaseType		EffectBaseType
TactileFlag	1	bslbf
tactileEffectFlag	1	bslbf
updateRateFlag	1	bslbf
if(TactileFlag) {		
SizeOfIntensityRow	4	uimsbf
SizeOfIntensityColumn	16	uimsbf
for(k=0;k<(SizeOfIntensityRow* SizeOfIntensityColumn);k++) {		
ArrayInstensity[k]	32	fsfb
}		
}		
else {		
TactileVideo	See ISO 10646	UTF-8
}		
if(tactileEffectFlag){		
tactileEffect	3	bslbf
}		
if(updateRateFlag) {		
updateRate	16	uimsbf
}		
}		

TactileResourceType {	*Number of bits*	*Mnemonic*
typeLength		vluimsbf5
type	typeLength *8	bslbf
ref		UTF-8
}		

5.17.4 Semantics

Semantics of the TactileType:

Name	*Definition*
TactileType	Tool for describing a tactile effect. Tactile effects can provide vibrations, pressures, temperature, etc, directly onto some areas of human skin through many types of actuators such as vibration motors, air-jets, piezo-actuators, thermal actuators. A tactile effect may effectively be represented by an ArrayIntensity or by a TactileVideo, all of which can be composed of m by n matrix that is mapped to m by n actuators in a tactile device. A Tactile Video is defined as a grayscale video formed with m-by-n pixels matched to the m-by-n tactile actuator array.
tactileFlag	This field, which is only present in the binary representation, specifies the choice of the tectile effect source. If it is 1 then the ArrayIntensity is present, otherwise the TactileVideo is present.
ArrayIntensity	Describes intensities in terms of physical quantities for all elements of m by n matrix of the tactile actuators. For temperature tactile effect, for example, intensity is specified in the unit of Celsius. For vibration tactile effect, intensity is specified in the unit of mm (amplitude). For pressure tactile effect, intensity is specified in the unit of Newton/mm^2.
TactileResource	Describes intensities in terms of a resource.
tactileEffectFlag	This field, which is only present in the binary representation, indicates the presence of the tactileEffect attribute. If it is 1 then the tactileEffect attribute is present, otherwise the tactileEffect attribute is not present.
tactileEffect	Describes the tactile effect to use. A CS that may be used for this purpose is the TactileEffectCS defined in Annex A of ISO/IEC 23005-6. This refers the preferable tactile effects.
updateRateFlag	This field, which is only present in the binary representation, indicates the presence of the updateRate attribute. If it is 1 then the updateRate attribute is present, otherwise the updateRate attribute is not present.
updateRate	Describes a number of data update times per second.
SizeOfIntensityRow	Describes a row size of ArrayIntensity
SizeOfIntensityColumn	Describes a column size of ArrayIntensity
TactileVideo	Describes intensities in terms of grayscale(0-255) video of tactile information. This grayscale value(0-255) can be divided into several levels according to the number of levels that a device produces.

Semantics of the `TactileResourceType`:

Name	*Definition*
`TactileResourceType`	Tool for describing intensities in terms of a resource. EXAMPLE 1 A grayscale (0-255) video of tactile information. This grayscale value (0-255) can be divided into several levels according to the number of levels that a device produces. EXAMPLE 2 An audio file containing the waveform of a tactile vibration effect.
`typeLength`	This field, which is only present in the binary representation, describes the length of the `type` attribute.
`type`	Describes the data type of the resource as a concatenation of MIME media-type, sub-type, and parameters, as defined in IETF RFC 2045.
`ref`	Describes the URI value that identifies the resource, as defined in IETF RFC 3986.

5.17.5 Example

EXAMPLE 1 This example shows the description of Tactile effect with ArrayIntensity by the following semantics. A set of intensity is provided by a set of intensity values with updateRate. When a 3-by-3 array of data is given, these data are mapped to tactile devices (note that if a device is formed with i-by-j array, 3-by-3 array of data is resized (upsizing/downsizing). The tactile effect is activated at si:pts="5" and the preferable tactile effect is pressure.

```
<sedl:DescriptionMetadata>
 <sedl:ClassificationSchemeAlias alias="TACTILEEFFECT" href="urn:mpeg:mpeg-
v:01-SI-TactileEffectCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:TactileType" updateRate="2"
tactileEffect=":TACTILEEFFECT:pressure" si:pts="5">
    <sev:ArrayIntensity mpeg7:dim="3 3">250 250 250 0 0 0 150 120 150
</sev:ArrayIntensity>
</sedl:Effect>
```

EXAMPLE 2 This example shows the description of a Tactile effect with TactileResource by the following semantics. A set of intensity is provided by a tactile video from the link "`http://www.haptic.kr/tactile/sample1.mp4`", and 30 frames of greyscale tactile information are given per second. The tactile effect is activated at si:pts="1" and the preferable tactile effect is vibration.

```
<sedl:DescriptionMetadata>
 <sedl:ClassificationSchemeAlias alias="TACTILEEFFECT" href="urn:mpeg:mpeg-
v:01-SI-TactileEffectCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:TactileType" updateRate="30"
tactileEffect=":TACTILEEFFECT:vibration" si:pts="1">
  <sev:TactileResource type="video/mp4"
ref="http://www.haptic.kr/tactile/sample1.mp4"/>
</sedl:Effect>
```

This tactile resource refers a greyscale video including intensity information for array type tactile effect. The resolution of the tactile resource is relatively low and intended to match the physical resolution (the number of individual output devices, or tactors) of a tactile display device. The intensity of each grayscale value determines the magnitude of the cue to be delivered to each tactor. Any resolution differences between the stored video and that supported by a viewer's tactile device can be resolved through basic image resizing operations at viewing time.

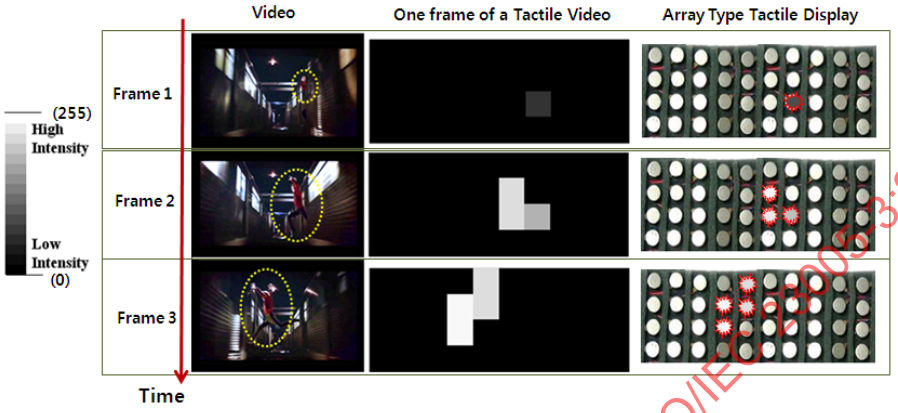


Figure 26 — Excerpt of example tactile resource

Figure 26 shows an example of one frame of a 10-by-4 pixel greyscale tactile video mapped to an equivalent 10-by-4 tactile display, composed of a uniform grid of vibrating motors and the marked vibrators are activated at the same time with different intensities according to the grayscale tactile video. In this Figure, lighter pixels correspond to more intense tactile cues.

EXAMPLE 3 This example shows the description of a Tactile effect using an audio tactile resource by the following semantics. The tactile effect is activated at si:pts="5" and the preferable tactile effect is vibration. The location is specified to be near the human heart region, on the front of the chest. The vibration effect is created based on a tactile resource which is an MP3 format audio file, stored in the same parent location (URL) as the SEDL script itself. The duration and speed of the tactile effect are entirely derived from the duration and default speed of the audio file.

```
<sedl:DescriptionMetadata>
 <sedl:ClassificationSchemeAlias alias="TACTILEEFFECT"
     href="urn:mpeg:mpeg-v:01-SI-TactileEffectCS-NS"/>
 <sedl:ClassificationSchemeAlias alias="LCS"
     href="urn:mpeg:mpeg-v:01-SI-LocationCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:TactileType"
tactileEffect=":TACTILEEFFECT:vibration"
     si:pts="5" location=":LCS:centerleft:middle:front">
  <sev:TactileResource type="audio/mpeg" ref="heartbeat1.mp3"/>
</sedl:Effect>
```


5.18 TactilePattern effect

5.18.1 Introduction

This Subclause specifies syntax and semantics of a tactile pattern effect.