INTERNATIONAL STANDARD

ISO/IEC 23005-4

Third edition
2016-03-15

Information technology — Media context and control —

Part 4: Virtual world object characteristics

Technologies de l'information — Contrôle et contexte de supports — Partie 4: Caractéristiques d'objet du monde virtuel

Reference number
ISO/IEC 23005-4:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This third edition cancels and replaces the second edition (ISO/IEC 23005-4:2013), 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

This International Standard provides an architecture and specifies associated information representations to enable interoperability between virtual worlds, e.g. digital content provider of a virtual world, gaming (serious), simulation, DVD, and 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 and navigation and the 3Cs 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.

1) 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.

2) 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 permit users to have unique experiences.

Although realistic experiences have been achieved via devices such as 3D audio/visual devices, it is hard to realize sensory effects only with 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 includes 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.

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 E.

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 E. ISO and the IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Media context and control —

Part 4:
Virtual world object characteristics

1 Scope

This part of ISO/IEC 23005 specifies syntax and semantics of description schemes and descriptors used to characterize a virtual world object related metadata, making it possible to migrate a virtual world object (or only its characteristics) from one virtual world to another and to control a virtual world object in a virtual world by real world devices.

The system architecture of this International Standard 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 R→V/V→R Adaptation and as an exchangeable information format to support interoperability between the virtual worlds, as defined in ISO/IEC 23005-1, is specified in this part of ISO/IEC 23005.

NOTE The actual R→V/V→R Adaptation is deliberately informative and left open for industry competition.

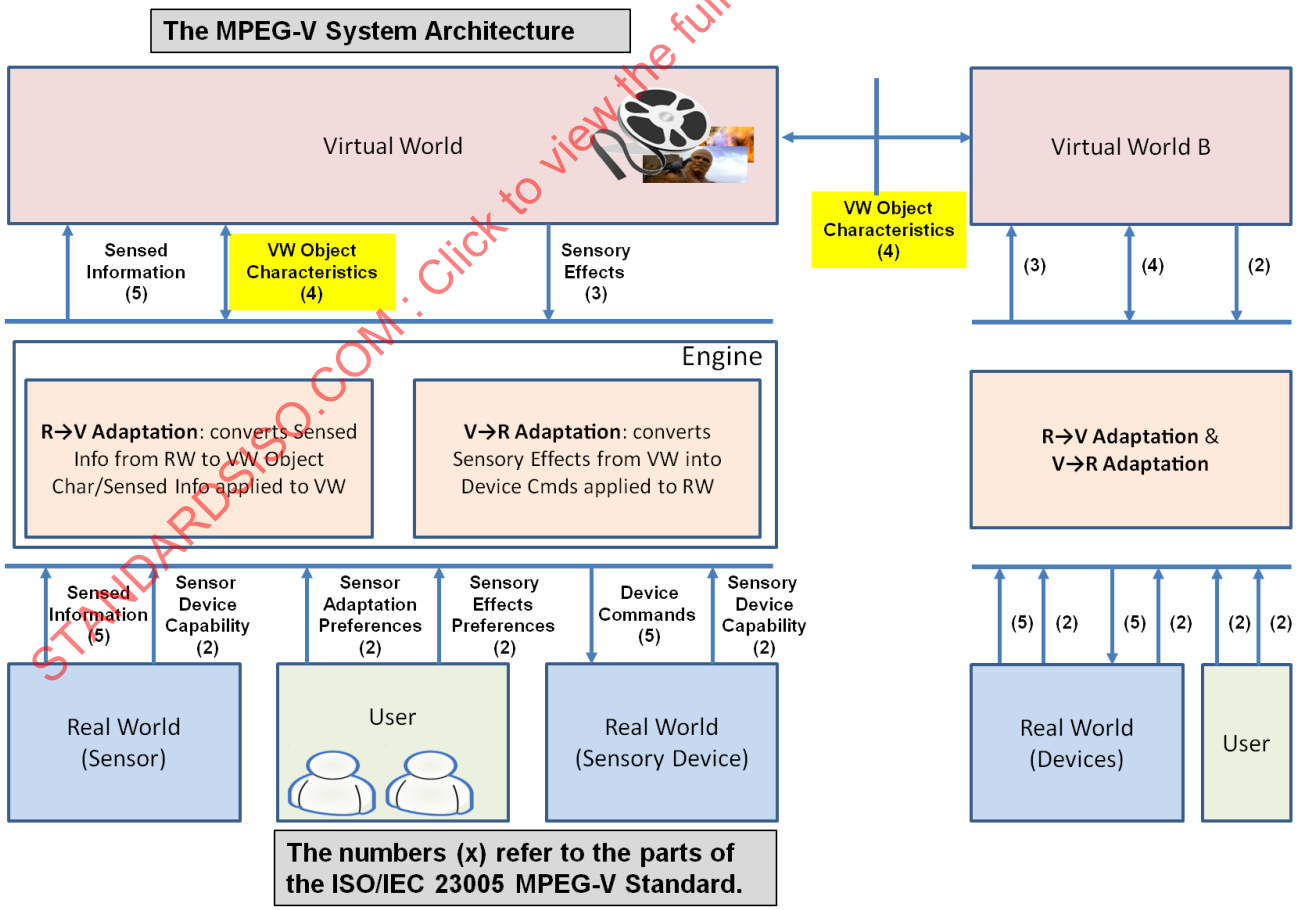


Figure 1 — System architecture

2 Normative references

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

ISO/IEC 15938-5:2003, *Information technology — Multimedia content description interface — Part 5: Multimedia description schemes*

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

ISO/IEC 23005-6, *Information technology — Media context and control — Part 6: Common types and tools*

3 Terms, definitions, abbreviated terms and prefixes

3.1 Terms and definitions

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

3.1.1
avatar
entity that can be used as a (visual) representation of the user inside the virtual environments

EXAMPLE A player's representation in the video game and human or fantastic representations of a person's self in non-gaming online worlds.

3.1.2
avatar metadata
defines the description schemes and descriptors to represent *avatars* (3.1.1)

3.1.3
Extensible Markup Language
set of rules for encoding documents in machine-readable form

3.1.4
Rights expression language
machine-readable language that declares rights and permissions

3.1.5
Uniform Resource Identifier
compact string of characters for identifying an abstract or physical resource

3.1.6
Uniform Resource Locator
compact string representation for a resource available via the Internet

3.1.7
virtual object
entity that is any (visual) object except for avatars in the virtual environment

3.1.8
virtual object metadata
defines the description schemes and descriptors to represent *virtual objects* (3.1.7)

3.1.9
virtual world object
entity that includes avatars and virtual objects in the virtual world

3.1.10
virtual world object metadata
defines the description schemes and descriptors to represent *virtual world objects* (3.1.9)

3.2 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

MPEG-21	multimedia framework (ISO/IEC 21000-5)
MPEG-7	multimedia content description interface (ISO/IEC 15938-5)
REL	rights expression language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	Extensible Markup Language

3.3 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 part of ISO/IEC 23005 to refer to an element of the `http://www.w3.org/2001/XMLSchema-instance` namespace.

"`xml:`" and "`xmlns:`" are normative prefixes defined in Reference [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 Table 1.

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

Prefix	Corresponding namespace
Ct	urn:mpeg:mpeg-v:2016:01-CT-NS
Sedl	urn:mpeg:mpeg-v:2016:01-SEDL-NS
Sev	urn:mpeg:mpeg-v:2016:01-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

4 Virtual world object metadata

4.1 General

A specificity of Virtual Environments (VEs) with respect to other multimedia applications consists in the representation of virtual world objects inside the environment. The "virtual world object" can be classified into

two types: avatars and virtual objects. An avatar can be used as a (visual) representation of the user inside the environment. These virtual world objects serve different purposes, namely:

— characterize various kinds of objects within the VE;

— provide an interaction with the VE.

In general, creating an object is a time-consuming task. Even though some components of the object may be related to the virtual environment (e.g. the avatar wearing a medieval suite in a contemporary style VE may be inappropriate), there is a real need of being able to create the object once and import/use it in different VEs. To serve the latter purpose, it should be possible to control the object from external applications (e.g. the emotions one avatar exposes in the VE can be obtained by processing the associated user's physiological sensors).The current standard proposes an XML Schema, called Virtual World Object Characteristics XSD, for describing an object by considering three main requirements.

— It should be possible to easily create importers/exporters from various VEs implementations.

— It should be easy to control an object within a VE.

— It should be possible to modify a proprietary template (specific to the virtual world) of the object by using data contained in Virtual World Object Characteristics file.

In detail, once the object is created possibly by an authoring tool specific to a VW, it can be used in any other VWs. In case of avatars, a user can have one's own unique presentation inside all VWs, like in real life. He can change and upgrade his avatar, i.e. "virtual himself" in one VW and then all the updated properties will be reflected in all the other VWs. The avatar itself contains representation and animation features but also higher level semantic information. However, each VW may have its own internal structure for handling avatars. ISO/IEC 23005 (MPEG-V) is not imposing any specific constraints on the internal structure of representing data by the VW, but proposes a descriptive format able to drive the transformation of a template or a creation from scratch of an avatar compliant with the VW. All the associated characteristics of the avatar (including the associated motion) can be exported from a VW and then imported to another VW. Similarly, any virtual object created by a user can also be exchangeable between VWs by exporting and importing the associated characteristics of the object. In case of interfacing between virtual worlds and the real world, the sensed real world information will be processed to obtain the meaningful data which can be used as control parameters on the associated characteristics of the object in the VW. As for avatar, the captured gesture of a user can be used to control the gesture of the avatar in the VW by updating the associated characteristics of the avatar. Similarly, the avatar motions created in the virtual world can be mapped onto a real robot for the use in dangerous areas, the maintenance tasks, the support for disabled and/or elderly people, and the like.

The proposed schema deals only with metadata and does not include representation of the geometry, sound, scent, animation or texture. To represent the latter, references to media resources are used. To provide a full interoperable solution, it may be combined with ISO/IEC 14496-16 (MPEG-4 Part 16) which includes a framework for defining and animating avatars and/or ISO/IEC 14496-11 (MPEG-4 Part 11) which includes a framework for defining graphical assets.

There is a base type of attributes and characteristics of the virtual world objects which is shared by both avatars and virtual objects.

The base type of the virtual world object characteristics is composed of following type of data:

— **identity**: contains identification descriptors;

— **sound**: contains sound resources and the related properties;

— **scent**: contains scent resources and the related properties;

— **control**: contains a set of descriptors for controlling motion features of an object such as translation, orientation and scaling;

— **event**: contains a set of descriptors providing input events from a mouse, keyboard and etc.;

— **behaviour model**: contains a set of descriptors defining the behavior information of the object according to input events;

— **id**: contains a unique identifier for identifying individual virtual world object information.

The virtual world object base type is inherited to both avatar metadata and virtual object metadata to extend the specific aspects of each of metadata.

4.2 Schema wrapper conventions

The syntax defined in this Clause assumes the following Schema Wrapper to form a valid XML schema document.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004" xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-
NS" xmlns:mpegvct="urn:mpeg:mpeg-v:2016:01-CT-NS" xmlns:vwoc="urn:mpeg:mpeg-
v:2016:01-VWOC-NS" targetNamespace="urn:mpeg:mpeg-v:2016:01-VWOC-NS"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="ISO/IEC 23005-4" id="MPEG-V-VWOC.xsd">
   <!-- ################################################ -->
   <!--  Import of reference schema                      -->
   <!-- ################################################ -->
   <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-REL-R-NS"
schemaLocation="http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-
21_schema_files/rel-r/rel-r.xsd"/>
   <import namespace="urn:mpeg:mpeg-v:2016:01-CT-NS"
schemaLocation="http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-
V_schema_files/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>
```

4.3 Root element and top-level tools

4.3.1 General

This subclause specifies the root element and the top-level tools which can follow root element in virtual world object characteristics information. The root element is the only element which can appear as the topmost element when the world object characteristics information specified in this part of ISO/IEC 23005 is instantiated. The top-level tools are defined as the elements which are allowed to appear as the topmost element within the root element.

4.3.2 XML representation syntax

```
<!-- ################################################ -->
<!--  Declaration of Root Element                     -->
<!-- ################################################ -->
<element name="VWOCInfo" type="vwoc:VWOCInfoType"/>

<complexType name="VWOCInfoType">
   <sequence>
      <element name="AvatarList" type="vwoc:AvatarListType" minOccurs="0"/>
      <element name="VirtualObjectList" type="vwoc:VirtualObjectListType"
minOccurs="0"/>
   </sequence>
```

```
</complexType>

<complexType name="AvatarListType">
    <sequence>
        <element name="Avatar" type="vwoc:AvatarBaseType" maxOccurs="unbounded"/>
    </sequence>
</complexType>

<complexType name="VirtualObjectListType">
    <sequence>
        <element name="VirtualObject" type="vwoc:VirtualObjectBaseType"
maxOccurs="unbounded"/>
    </sequence>
</complexType>
```

4.3.3 Binary representation syntax

	Number of bits	***Mnemonic***
VWOCInfo		VWOCInfoType
VWOCInfoType{		
AvatarListFlag	1	bslbf
VritualObjectListFlag	1	bslbf
if(AvatarListFlag){		
AvatarList		AvatarListType
}		
if(VirtualObjectListFlag){		
VirtualObjectList		VirtualObjectListType
}		
AvatarListType{		
NumAvatarType		vluimsbf5
for(k=0;k< NumAvatarType;k++){		
IndividualAvatarType	8	bslbf
Avatar		AvatarBaseType
}		
}		
VirtualObjectListType{		

NumVirtualObjectType		vluimsbf5
for(k=0;k< NumVirtualObjectType;k++){		
IndividualVirtualObjectType	16	bslbf
VirtualObject		VirtualObjectBaseType
}		
}		

4.3.4 Semantics

Name	*Description*
VWOCInfo	The root element that serves as the topmost element in the virtual world object characteristics description.
VWOCInfoType	The root type provides basic structure that the virtual world object characteristics information description should follow through the root element.
AvatarListFlag	This field, which is only present in the binary representation, signals the presence of the AvatarList element. "1" means that the element shall be used. "0" means that the element shall not be used.
VirtualObjectListFlag	This field, which is only present in the binary representation, signals the presence of the VirtualObjectList element. "1" means that the element shall be used. "0" means that the element shall not be used.
AvatarList	Optional wrapper element that serves as the placeholder for the list of avatar characteristics information.
VirtualObjectList	Optional wrapper element that serves as the placeholder for the list of virtual object characteristics information.
AvatarListType	Wrapper element type which allows multiple occurrences of avatar characteristics information.
NumAvatarType	This field, which is only present in the binary representation, specifies the number of Avatar information contained in the AvatarListType.
Avatar	Specifies the description of avatar characteristics information.
AvatarBaseType	AvatarBaseType is a type providing a characteristic description of an individual avatar.
IndividualAvatarType	This field, which is only presented in the binary representation, specifies the types of each avatar. *Individual Avatar Type* / *Binary representation for avatar type (8 bits)* AvatarType — 00000000 MakeupAvatarType — 00000001 Reserved — 00000010-11111111
VirtualObjectListType	Wrapper element type which allows multiple occurrences of virtual object characteristics information.
NumVirtualObjectType	This field, which is only present in the binary representation, specifies the number of virtual object information contained in the virtual object list type.
VirtualObject	Specifies the description of virtual object characteristics information.
VirtualObjectBaseType	VirtualObjectBaseType is a type providing a characteristic description of an individual virtual object.
IndividualVirtualObjectType	This field, which is only presented in the binary representation, specifies the types of each virtual object.

	Individual Virtual Object Type	*Binary representation for virtual object type (16 bits)*	
	VirtualObjectType	0000hex	
	Reserved	0001hex-FFFFhex	

4.3.5 Examples

The following shows two use cases of VWOCInfo element, which are for listing avatar characteristics information and for listing virtual object characteristics information.

The first example shows the case when the VWOCInfo is used for AvatarList.

```
<vwoc:VWOCInfo xsi:schemaLocation="urn:mpeg:mpeg-v:2016:01-VWOC-NS
VWOCSchema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpegvct="urn:mpeg:mpeg-v:2012:01-CT-NS" xmlns:vwoc="urn:mpeg:mpeg-
v:2016:01-VWOC-NS" xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
    <vwoc:AvatarList>
        <vwoc:Avatar xsi:type="vwoc:AvatarType" id="ID_1" gender="male">
            . . .
        </vwoc:Avatar>
    </vwoc:AvatarList>
</vwoc:VWOCInfo>
```

The second example shows the case when the VWOCInfo is used for VirtualObjectList.

```
<vwoc:VWOCInfo xsi:schemaLocation="urn:mpeg:mpeg-v:2016:01-VWOC-NS
VWOCSchema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpegvct="urn:mpeg:mpeg-v:2012:01-CT-NS" xmlns:vwoc="urn:mpeg:mpeg-
v:2016:01-VWOC-NS" xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
    <vwoc:VirtualObjectList>
        <vwoc:VirtualObject xsi:type="vwoc:VirtualObjectType" id="ID_80">
            . . .
        </vwoc:VirtualObject>
    </vwoc:VirtualObjectList>
</vwoc:VWOCInfo>
```

Note that these examples are only showing a part of the complete XML description to show the use of the root element, VWOCInfo, with the AvatarList and the VirtualObjectList.

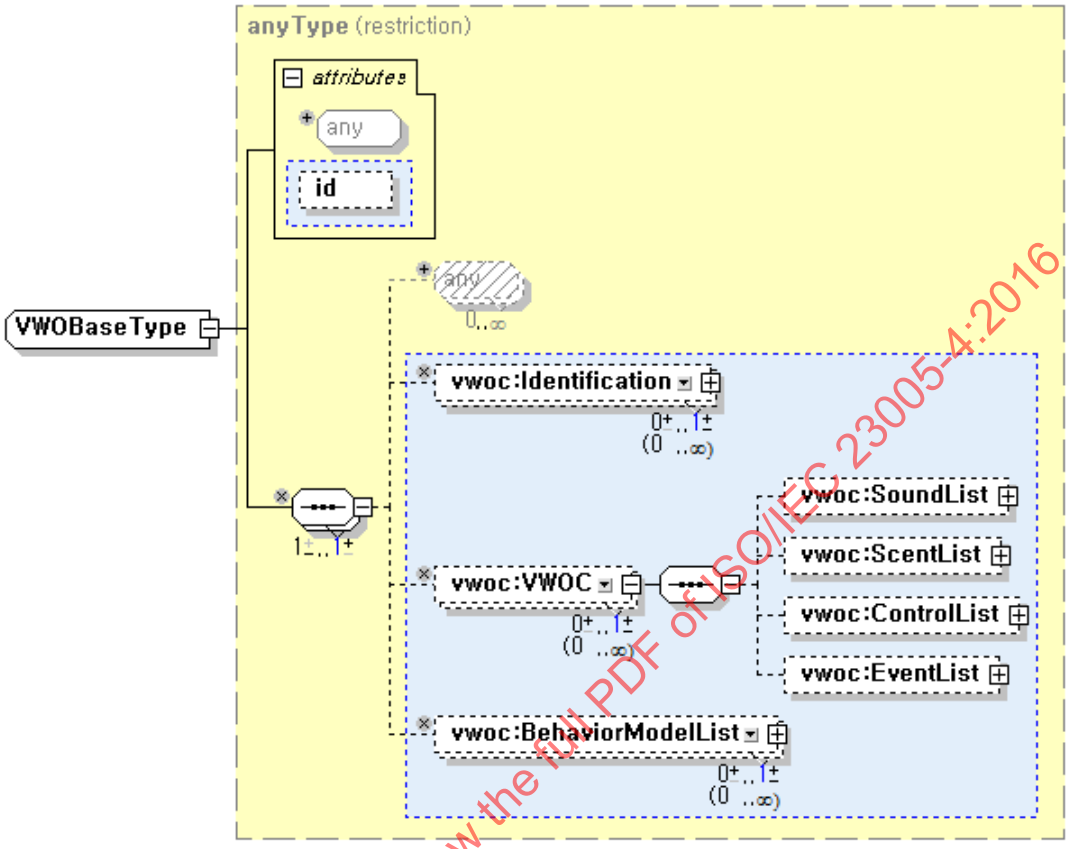
4.4 Virtual world object base type

4.4.1 General

This subclause defines a complex type of VWOBaseType, which the avatar characteristics information and virtual object characteristics information should inherit.

4.4.2 XML representation syntax

Diagram



Source

```
<complexType name="VWOBaseType" abstract="true">
  <complexContent>
    <restriction base="anyType">
      <sequence>
        <element name="Identification" type="vwoc:IdentificationType" minOccurs="0"/>
        <element name="Description" type="string" minOccurs="0"/>
        <element name="VWOC" minOccurs="0">
          <complexType>
            <sequence>
              <element name="SoundList" type="vwoc:VWOSoundListType" minOccurs="0"/>
              <element name="ScentList" type="vwoc:VWOScentListType" minOccurs="0"/>
              <element name="ControlList" type="vwoc:VWOControlListType" minOccurs="0"/>
              <element name="EventList" type="vwoc:VWOEventListType" minOccurs="0"/>
            </sequence>
          </complexType>
        </element>
        <element name="BehaviorModelList" type="vwoc:VWOBehaviorModelListType" minOccurs="0"/>
      </sequence>
      <attribute name="id" type="ID" use="optional"/>
    </restriction>
  </complexContent>
</complexType>

<complexType name="AvatarBaseType" abstract="true">
  <complexContent>
    <extension base="vwoc:VWOBaseType"/>
  </complexContent>
</complexType>

<complexType name="VirtualObjectBaseType" abstract="true">
  <complexContent>
    <extension base="vwoc:VWOBaseType"/>
  </complexContent>
</complexType>
```

4.4.3 Binary representation syntax

VWOBaseType{	***Number of bits***	***Mnemonic***
IdentificationFlag	1	Bslbf
DescriptionFlag	1	Bslbf
VWOCFlag	1	Bslbf
BehaviorModelListFlag	1	bslbf
IdFlag	1	bslbf
if(IdentificationFlag) {		
Identification		IdentificationType
}		
if(DescriptionFlag) {		
Description	See ISO 10646	UTF-8
}		
if(VWOCFlag) {		
SoundListFlag	1	bslbf
ScentListFlag	1	bslbf
ControlListFlag	1	bslbf
EventListFlag	1	bslbf
if(SoundListFlag) {		
SoundList		VWOSoundListType
}		
if(ScentListFlag) {		
ScentList		VWOScentListType
}		
if(ControlListFlag) {		
ControlList		VWOControlListType
}		
if(EventListFlag) {		

EventList		VWOEventListType
}		
}		
if(BehaviorModelListFlag) {		
BehaviorModelList		VWOBehaviorModelListType
}		
if(IdFlag) {		
id	See ISO 10646	UTF-8
}		
}		
AvatarBaseType {		
VWOBase		VWOBaseType
}		
VirtualObjectBaseType {		
VWOBase		VWOBaseType
}		

4.4.4 Semantics

Name	*Description*
`VWOBaseType`	The base type that describes common attributes and elements in both avatars and virtual objects.
`DescriptionFlag`	This field, which is only presented in the binary representation, signals the presence of the description element. “1” means that the element shall be used. “0” means that the element shall not be used.
`VWOCFlag`	This field, which is only presented in the binary representation, signals the presence of the `VWOC` element which contains sound, scent, control and event lists. “1” means that the element shall be used. “0” means that the element shall not be used.
`IdFlag`	This field, which is only presented in the binary representation, signals the presence of the `id` attribute. “1” means that the element shall be used. “0” means that the element shall not be used.
`IdentificationFlag`	This field, which is only present in the binary representation, signals the presence of the `Identification` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.

SoundListFlag	This field, which is only present in the binary representation, signals the presence of the Sound element list. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ScentListFlag	This field, which is only present in the binary representation, signals the presence of the Scent element list. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ControlListFlag	This field, which is only present in the binary representation, signals the presence of the Control element list. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EventListFlag	This field, which is only present in the binary representation, signals the presence of the Event element list. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BehaviorModelListFlag	This field, which is only present in the binary representation, signals the presence of the BehaviorModel element list. ”1” means that the element shall be used. ”0” means that the element shall not be used.
Identification	Describes the identification of the virtual world object.
Description	Contains the description of the virtual world object.
VWOC	Describes a set of characteristics of the virtual world objects.
SoundList	Describes a list of the sound effects associated to the virtual world object.
ScentList	Describes a list of the scent effects associated to the virtual world object.
ControlList	Describes a list of the controls associated to the virtual world object.
EventList	Describes a list of the input events associated to the virtual world object.
BehaviorModelList	Describes a list of the behaviour models associated to the virtual world object.
id	Unique identifier for identifying individual virtual world object information.
AvatarBaseType	A type providing a characteristic description of an individual avatar.
VirtualObjectBaseType	A type providing a characteristic description of an individual virtual object.

4.4.5 Examples

```
<vwoc:VWOCInfo xsi:schemaLocation="urn:mpeg:mpeg-v:2016:01-VWOC-NS
VWOCSchema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpegvct="urn:mpeg:mpeg-v:2012:01-CT-NS" xmlns:vwoc="urn:mpeg:mpeg-
v:2016:01-VWOC-NS" xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
    <vwoc:AvatarList>
        <vwoc:Avatar xsi:type="vwoc:AvatarType" id="AVATARID_1" gender="male">
            <vwoc:VWOC>
                <vwoc:SoundList>
                    <vwoc:Sound loop="1" soundID="SOUNDID_10" duration="10"
intensity="3" name="BurpSound">
                    <vwoc:ResourcesURL>http://www.BurpSound.info</vwoc:ResourcesURL>
                    </vwoc:Sound>
                </vwoc:SoundList>
                <vwoc:ScentList>
                    <vwoc:Scent loop="2" duration="1" intensity="3"
name="BurpingScent" scentID="SCENTID_11">
                        <vwoc:ResourcesURL>http://www.Burp.info</vwoc:ResourcesURL>
                    </vwoc:Scent>
                </vwoc:ScentList>
                <vwoc:ControlList>
                    <vwoc:Control controlID="CTRLID_12">
                        <vwoc:MotionFeatureControl>
```

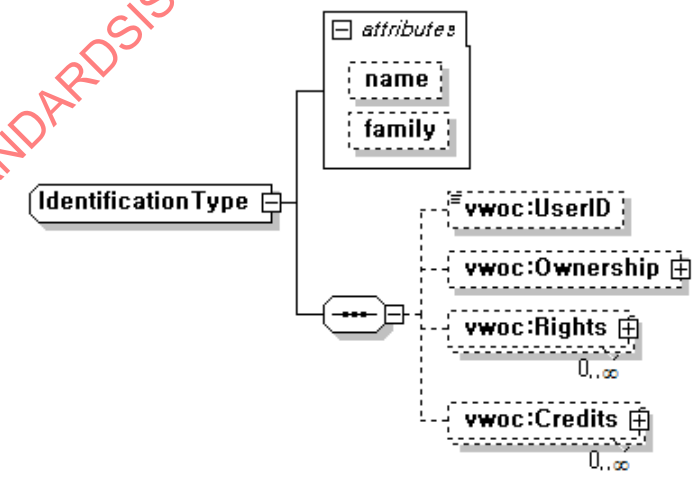


```
                        <vwoc:Position>
                            <mpegvct:X>1</mpegvct:X>
                            <mpegvct:Y>1</mpegvct:Y>
                            <mpegvct:Z>10</mpegvct:Z>
                        </vwoc:Position>
                        <vwoc:Orientation>
                            <mpegvct:X>0</mpegvct:X>
                            <mpegvct:Y>0</mpegvct:Y>
                            <mpegvct:Z>0</mpegvct:Z>
                        </vwoc:Orientation>
                        <vwoc:ScaleFactor>
                            <mpegvct:X>1</mpegvct:X>
                            <mpegvct:Y>1</mpegvct:Y>
                            <mpegvct:Z>3</mpegvct:Z>
                        </vwoc:ScaleFactor>
                    </vwoc:MotionFeatureControl>
                </vwoc:Control>
            </vwoc:ControlList>
            <vwoc:EventList>
                <vwoc:Event eventID="ID_13">
                    <vwoc:Mouse>urn:mpeg:mpeg-v:01-VWOC-MouseEventCS-NS:click
</vwoc:Mouse>
                </vwoc:Event>
            </vwoc:EventList>
        </vwoc:VWOC>
        <vwoc:BehaviorModelList>
            <vwoc:BehaviorModel>
                <vwoc:BehaviorInput eventIDRef="ID_13"/>
                <vwoc:BehaviorOutput controlIDRefs="CTRLID_12"
scentIDRefs="SCENTID_11" soundIDRefs="SOUNDID_10"/>
            </vwoc:BehaviorModel>
        </vwoc:BehaviorModelList>
    </vwoc:Avatar>
</vwoc:AvatarList>
</vwoc:VWOCInfo>
```


4.4.6 IdentificationType

4.4.6.1 XML representation syntax

Diagram



Source	<complexType name="IdentificationType"> <sequence> <element name="UserID" type="anyURI" minOccurs="0"/> <element name="Ownership" type="mpeg7:AgentType" minOccurs="0"/> <element name="Rights" type="r:License" minOccurs="0" maxOccurs="unbounded"/> <element name="Credits" type="mpeg7:AgentType" minOccurs="0" maxOccurs="unbounded"/> </sequence> <attribute name="name" type="string" use="optional"/> <attribute name="family" type="string" use="optional"/> </complexType>

4.4.6.2 Binary representation syntax

IdentificationType {	***Number of bits***	***Mnemonic***
UserIDFlag	1	bslbf
OwnershipFlag	1	bslbf
RightsFlag	1	bslbf
CreditsFlag	1	bslbf
nameFlag	1	bslbf
familyFlag	1	bslbf
if(UserIDFlag) {		
UserID	See ISO 10646	UTF-8
}		
if(OwnershipFlag) {		
Ownership		AgentType
}		
if(RightsFlag) {		
NumRights		vluimsbf5
for(k=0; k< NumRights; k++){		
Rights[k]	See ISO/IEC 21000-16:2005	LicenseType
}		
}		
if(CreditsFlag) {		
NumCredits		vluimsbf5
for(k=0; k< NumCredits; k++){		

Credits[k]		AgentType
}		
}		
if(nameFlag) {		
name	See ISO 10646	UTF-8
}		
if(familyFlag) {		
family	See ISO 10646	UTF-8
}		
}		
AgentType{	***Number of bits***	***Mnemonic***
mpeg7:AgentType		UTF-8
}		
LicenseType{	***Number of bits***	***Mnemonic***
r:LicenseType		UTF-8
}		

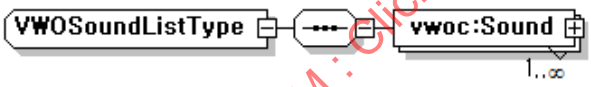
4.4.6.3 Semantics

Name	*Definition*
`IdentificationType`	Describes the identification of a virtual world object.
`UserIDFlag`	This field, which is only present in the binary representation, signals the presence of the `UserID` element. "1" means that the element shall be used. "0" means that the element shall not be used.
`OwnershipFlag`	This field, which is only present in the binary representation, signals the presence of the `Ownership` element. "1" means that the element shall be used. "0" means that the element shall not be used.
`RightsFlag`	This field, which is only presented in the binary representation, signals the presence of the `rights` element. "1" means that the element shall be used. "0" means that the element shall not be used.
`CreditsFlag`	This field, which is only presented in the binary representation, signals the presence of the `credits` element. "1" means that the element shall be used. "0" means that the element shall not be used.
`nameFlag`	This field, which is only present in the binary representation, signals the presence of the `name` attribute. "1" means that the element shall be

	used. ”0” means that the element shall not be used.
familyFlag	This field, which is only present in the binary representation, signals the presence of the family attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
UserID	Contains the user identification associated to the virtual world object
Ownership	Describes the ownership of the virtual world object which shall be based on the type “AgentType” defined in ISO/IEC 15938-5:2003, 7.4.2. In the binary representation, the “AgentType” shall be encoded by UTF-8.
NumRights	This field, which is only present in the binary representation, specifies the number of rights information.
Rights	Describes the rights of the virtual world object which shall be based on the type “LicenseType” defined in ISO/IEC 21000-5. In the binary representation, the “LicenseType” shall be encoded by UTF-8.
NumCredits	This field, which is only present in the binary representation, specifies the number of credits information.
Credits	Describes the contributors of the virtual object in chronological order which shall be based on the type “AgentType” defined in ISO/IEC 15938-5:2003, 7.4.2. In the binary representation, the “AgentType” shall be encoded by UTF-8. Note: The first listed credit describes an original author of a virtual world object. The subsequent credits represent the list of the contributors of the virtual world object chronologically.
name	Describes the name of the virtual world object.
family	Describes the relationship with other virtual world objects.

4.4.7 VWOSoundListType

4.4.7.1 XML representation syntax

Diagram	
Source	<complexType name="VWOSoundListType"> <sequence> <element name="Sound" type="vwoc:VWOSoundType" maxOccurs="unbounded"/> </sequence> </complexType>

4.4.7.2 Binary representation syntax

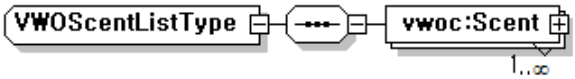
VWOSoundListType {	***Number of bits***	***Mnemonic***
NumVWOSoundType		vluimsbf5
for(k=0; k< NumVWOSoundType; k++){		
Sound[k]		VWOSoundType
}		
}		

4.4.7.3 Semantics

Name	*Definition*
VWOSoundListType	Wrapper element type which allows multiple occurrences of sound effects associated to the virtual world object.
NumVWOSoundType	This field, which is only present in the binary representation, specifies the number of Sound information contained in the sound list type.
Sound	Describes a sound effect associated to the virtual world object.

4.4.8 VWOScentListType

4.4.8.1 XML representation syntax

Diagram	
Source	<complexType name="VWOScentListType"> <sequence> <element name="Scent" type="vwoc:VWOScentType" maxOccurs="unbounded"/> </sequence> </complexType>

4.4.8.2 Binary representation syntax

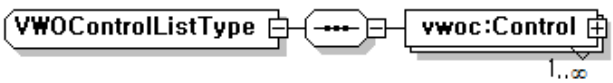
VWOScentListType {	***Number of bits***	***Mnemonic***
NumVWOScentType		vluimsbf5
for(k=0; k< NumVWOScentType; k++){		
Scent[k]		VWOScentType
}		
}		

4.4.8.3 Semantics

Name	*Definition*
VWOScentListType	Wrapper element type which allows multiple occurrences of Scent effects associated to the virtual world object.
NumVWOScentType	This field, which is only present in the binary representation, specifies the number of Scent information contained in the scent list type.
Scent	Describes a scent effect associated to the virtual world object.

4.4.9 VWOControlListType

4.4.9.1 XML representation syntax

Diagram	
Source	<complexType name="VWOControlListType"> <sequence>

	<element name="Control" type="vwoc:VWOControlType" maxOccurs="unbounded"/> </sequence> </complexType>

4.4.9.2 Binary representation syntax

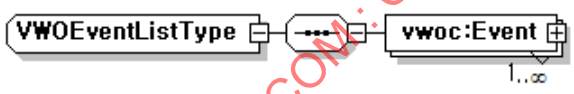
VWOControlListType {	*Number of bits*	*Mnemonic*
NumVWOControlType		vluimsbf5
for(k=0; k< NumVWOControlType; k++){		
Control[k]		VWOControlType
}		
}		

4.4.9.3 Semantics

Name	*Definition*
`VWOControlListType`	Wrapper element type which allows multiple occurrences of the controls associated to the virtual world object.
`NumVWOControlType`	This field, which is only present in the binary representation, specifies the number of `control` information contained in the Control list type.
`Control`	Describes a control associated to the virtual world object.

4.4.10 VWOEventListType

4.4.10.1 XML representation syntax

Diagram	
Source	<complexType name="VWOEventListType"> <sequence> <element name="Event" type="vwoc:VWOEventType" maxOccurs="unbounded"/> </sequence> </complexType>

4.4.10.2 Binary representation syntax

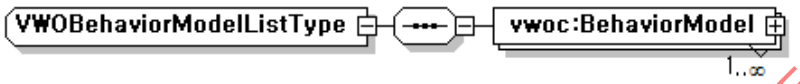
VWOEventListType {	*Number of bits*	*Mnemonic*
NumVWOEventType		vluimsbf5
for(k=0; k< NumVWOEventType; k++){		
Event[k]		VWOEventType
}		
}		

4.4.10.3 Semantics

Name	*Definition*
VWOEventListType	Wrapper element type which allows multiple occurrences of the input events associated to the virtual world object.
NumVWOEventType	This field, which is only present in the binary representation, specifies the number of Event information contained in the Event list type.
Event	Describes an input event associated to the virtual world object.

4.4.11 VWOBehaviorModelListType

4.4.11.1 XML representation syntax

Diagram	
Source	<complexType name="VWOBehaviorModelListType"> <sequence> <element name="BehaviorModel" type="vwoc:VWOBehaviorModelType" maxOccurs="unbounded"/> </sequence> </complexType>

4.4.11.2 Binary representation syntax

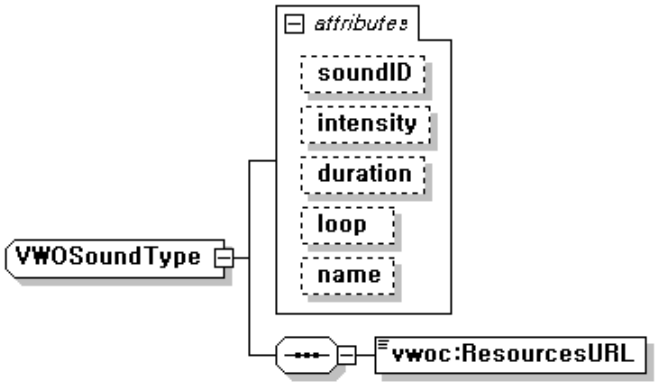
VWOBehaviorModelListType {	***Number of bits***	***Mnemonic***
NumVWOBehaviorModelType		vluimsbf5
for(k=0;k<NumVWOBehaviorModelType;k++){		
BehaviorModel[k]		VWOBehaviorModelType
}		
}		

4.4.11.3 Semantics

Name	*Definition*
VWOBehaviorModelListType	Wrapper element type which allows multiple occurrences of the behavior models associated to the virtual world object.
NumVWOBehaviorModelType	This field, which is only present in the binary representation, specifies the number of BehaviorModel information contained in the behavior model list type.
BehaviorModel	Describes a behavior model associated to the virtual world object.

4.4.12 VWOSoundType

4.4.12.1 XML representation syntax

Diagram	
Source	<complexType name="VWOSoundType"> <sequence> <element name="ResourcesURL" type="anyURI"/> </sequence> <attribute name="soundID" type="ID" use="optional"/> <attribute name="intensity" type="float" use="optional"/> <attribute name="duration" type="unsignedInt" use="optional"/> <attribute name="loop" type="unsignedInt" use="optional"/> <attribute name="name" type="string" use="optional"/> </complexType>

4.4.12.2 Binary representation syntax

VWOSoundType{	***Number of bits***	***Mnemonic***
SoundIDFlag	1	bslbf
IntensityFlag	1	bslbf
DurationFlag	1	bslbf
LoopFlag	1	bslbf
NameFlag	1	bslbf
ResourcesURL	See ISO 10646	UTF-8
if(SoundIDFlag) {		
soundID	See ISO 10646	UTF-8
}		
if(IntensityFlag) {		
intensity	32	fsbf
}		
if(DurationFlag) {		

duration	32	uimsbf
}		
if(LoopFlag) {		
loop	8	uimsbf
}		
if(NameFlag) {		
name	See ISO 10646	UTF-8
}		
}		

4.4.12.3 Semantics

Name	*Definition*
VWOSoundType	A type that contains the descriptions of a sound effect associated to the virtual world object.
SoundIDFlag	This field, which is only present in the binary representation, signals the presence of the ID attribute of the sound. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
IntensityFlag	This field, which is only present in the binary representation, signals the presence of the intensity attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
DurationFlag	This field, which is only present in the binary representation, signals the presence of the duration attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
LoopFlag	This field, which is only present in the binary representation, signals the presence of the loop attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
NameFlag	This field, which is only present in the binary representation, signals the presence of the name attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
SoundResources URL	Element that contains a link to sound file, usually MP4 file.
soundID	A unique identifier of the object sound.
intensity	The strength(volume) of the sound
duration	The length of time that the sound lasts. The default unit is ms.
loop	A playing option to describe the number of repetition (default value: 1, 0: indefinite repetition, 1:once, 2: twice, …, n: n times)
name	The name of the sound.

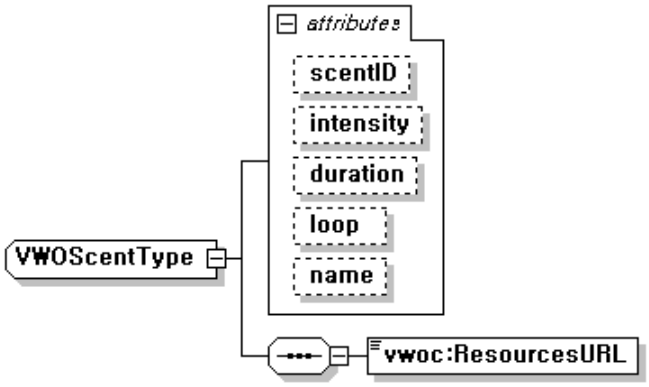
4.4.12.4 Examples

This example shows the description of the sound information associated to an object with the following semantics. The sound resource whose name is “BigAlarm” is saved at “http://sounddb.com/alarmsound_0001.wav” and the value of soundID, its identifier is “SoundID3” The length of the sound is 30 seconds. The sound shall be played with the volume of intensity = “50 %” repeatedly.

```
<vwoc:Sound loop="0" soundID="SoundID3" duration="30" intensity="0.5"
name="BigAlarm">
    <vwoc:ResourcesURL>http://sounddb.com/alarmsound_0001.wav</vwoc:ResourcesURL>
</vwoc:Sound>
```

4.4.13 VWOScentType

4.4.13.1 XML representation syntax

Diagram	
source	<complexType name="VWOScentType"> <sequence> <element name="ResourcesURL" type="anyURI"/> </sequence> <attribute name="scentID" type="ID" use="optional"/> <attribute name="intensity" type="float" use="optional"/> <attribute name="duration" type="unsignedInt" use="optional"/> <attribute name="loop" type="unsignedInt" use="optional"/> <attribute name="name" type="string" use="optional"/> </complexType>

4.4.13.2 Binary representation syntax

VWOScentType{	***Number of bits***	***Mnemonic***
ScentIDFlag	1	bslbf
IntensityFlag	1	bslbf
DurationFlag	1	bslbf
LoopFlag	1	bslbf
NameFlag	1	bslbf
ResourcesURL	See ISO 10646	UTF-8
if(ScentIDFlag) {		
scentID	See ISO 10646	UTF-8
}		
if(IntensityFlag) {		

intensity	32	fsbf
}		
if(DurationFlag) {		
duration	32	uimsbf
}		
if(LoopFlag) {		
loop	8	uimsbf
}		
if(NameFlag) {		
name	See ISO 10646	UTF-8
}		
}		

4.4.13.3 Semantics

Name	*Definition*
VWOScentType	A type that contains the descriptions of a scent effect associated to the virtual world object.
ScentIDFlag	This field, which is only present in the binary representation, signals the presence of the ID attribute of the scent. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
IntensityFlag	This field, which is only present in the binary representation, signals the presence of the intensity attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
DurationFlag	This field, which is only present in the binary representation, signals the presence of the duration attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
LoopFlag	This field, which is only present in the binary representation, signals the presence of the loop attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
NameFlag	This field, which is only present in the binary representation, signals the presence of the name attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
ScentResources URL	Element that contains a link to a scent file.
scentID	A unique identifier of the object scent.
intensity	The strength of the scent.
duration	The length of time that the scent lasts. The default unit is ms.
loop	A playing option to describe the number of repetition (default value: 1, 0: indefinite repetition, 1:once, 2: twice, …, n: n times)
name	The name of the scent.

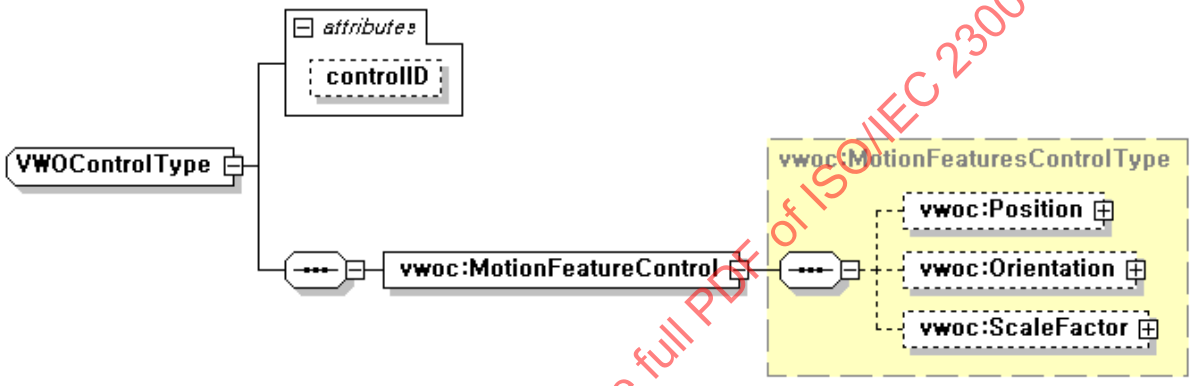
4.4.13.4 Examples

This example shows the description of the scent information associated to the object. The scent resource whose name is "rose" is saved at "http://scentdb.com/flower_0001.sct" and the value of scentID, its identifier is "ScentID5". The intensity shall be 20 % with duration of 20 seconds.

```
<vwoc:Scent duration="20" intensity="0.2" name="rose" scentID="ScentID5">
    <vwoc:ResourcesURL>http://scentdb.com/flower_0001.sct</vwoc:ResourcesURL>
</vwoc:Scent>
```

4.4.14 VWOControlType

4.4.14.1 XML representation syntax

Diagram	
source	<complexType name="VWOControlType"> <sequence> <element name="MotionFeatureControl" type="vwoc:MotionFeaturesControlType"/> </sequence> <attribute name="controlID" type="ID" use="optional"/> </complexType> <complexType name="MotionFeaturesControlType"> <sequence> <element name="Position" type="mpegvct:Float3DVectorType" minOccurs="0"/> <element name="Orientation" type="mpegvct:Float3DVectorType" minOccurs="0"/> <element name="ScaleFactor" type="mpegvct:Float3DVectorType" minOccurs="0"/> </sequence> </complexType>

4.4.15 Binary representation syntax

VWOControlType {	***Number of bits***	***Mnemonic***
ControlIDFlag	1	bslbf
MotionFeatureControl		MotionFeatureControlType
if(ControlIDFlag) {		
controlID	See ISO 10646	UTF-8
}		
}		
MotionFeaturesControlType{		

PositionFlag	1	bslbf
OrientationFlag	1	bslbf
ScaleFactorFlag	1	bslbf
if(PositionFlag) {		
Position		Float3DVectorType
}		
if(OrientationFlag) {		
Orientation		Float3DVectorType
}		
if(ScaleFactorFlag) {		
ScaleFactor		Float3DVectorType
}		
}		

4.4.15.1 Semantics

Name	*Definition*
`VWOControlType`	A type that contains the descriptions of a control associated to the virtual world object.
`ControlIDFlag`	This field, which is only present in the binary representation, signals the presence of the `ControlID` element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.

<table>
<tr><td>MotionFeatureControl</td><td>Set of elements that control position, orientation and scale of the virtual object.
<table>
<tr><th>Element</th><th>Information</th></tr>
<tr><td>MotionFeatureControlType</td><td>A type that provides three types of controls such as position control, orientation control, and scaling control.</td></tr>
<tr><td>PositionFlag</td><td>This field, which is only present in the binary representation, signals the presence of the Position element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.</td></tr>
<tr><td>OrientationFlag</td><td>This field, which is only present in the binary representation, signals the presence of the Orientation element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.</td></tr>
<tr><td>ScaleFactorFlag</td><td>This field, which is only present in the binary representation, signals the presence of the ScaleFactor element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.</td></tr>
<tr><td>Position</td><td>The position of the object in the scene with 3D floating point vector (x, y, z).</td></tr>
<tr><td>Orientation</td><td>The orientation of the object in the scene with 3D floating point vector as an Euler angle (yaw, pitch, roll).</td></tr>
<tr><td>ScaleFactor</td><td>The scale of the object in the scene expressed as 3D floating point vector (Sx, Sy, Sz).</td></tr>
</table></td></tr>
<tr><td>controlID</td><td>A unique identifier of the control.</td></tr>
</table>

NOTE 1 If two controllers are associated to the same object but on different parts of the object and if these parts exist hierarchical structures (parent and children relationship), then the controllers do perform the relative motion of the children. If the controllers are associated with the same part, the controller does the scaling or similar effects for the entire object.

NOTE 2 The reference coordinate system of this part is the right-handed coordinate system.

4.4.15.2 Examples

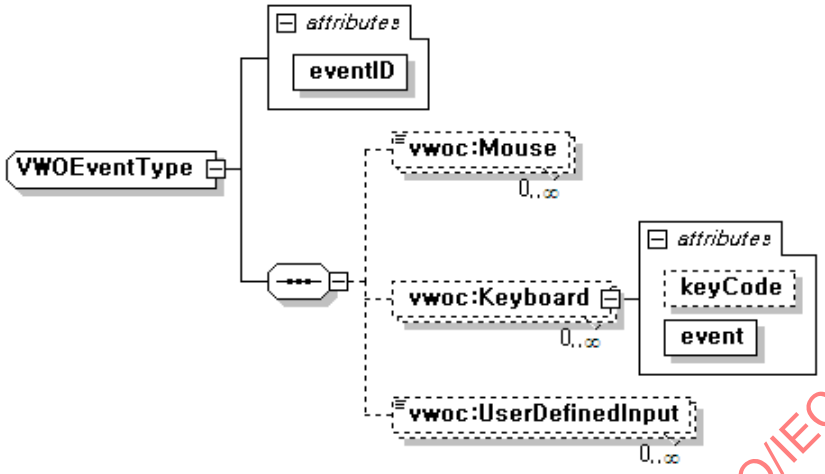
This example shows the description of object control information with the following semantics. The motion feature control of changing a position is given and its value of controlID, its identifier is “CtrlID7”. The object shall be positioned at X=”122,0”, Y=”150,0” and Z=”40,0”.

```
<vwoc:Control controlID="CtrlID7">
   <vwoc:MotionFeatureControl>
      <vwoc:Position>
         <mpegvct:X>122.0</mpegvct:X>
         <mpegvct:Y>150.0</mpegvct:Y>
         <mpegvct:Z>40.0</mpegvct:Z>
      </vwoc:Position>
   </vwoc:MotionFeatureControl>
</vwoc:Control>
```


4.4.16 VWOEventType

4.4.16.1 XML representation syntax

Diagram



Source

```
<complexType name="VWOEventType">
    <sequence>
        <element name="Mouse" type="mpeg7:termReferenceType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Keyboard" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <attribute name="keyCode" type="mpeg7:unsigned8" use="optional"/>
                <attribute name="event" use="required">
                    <simpleType>
                        <restriction base="string">
                            <enumeration value="pressed"/>
                            <enumeration value="clicked"/>
                            <enumeration value="released"/>
                        </restriction>
                    </simpleType>
                </attribute>
            </complexType>
        </element>
        <element name="UserDefinedInput" type="string" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="eventID" type="ID" use="required"/>
</complexType>
```

4.4.16.2 Binary representation syntax

VWOEventType {	*Number of bits*	*Mnemonic*
MouseFlag	1	bslbf
KeyboardFlag	1	bslbf
UserDefinedInputFlag	1	bslbf
if(MouseFlag) {		
NumOfMouse		vluimsbf5
for (k=0; k<NumOfMouse; k++) {		

Mouse[k]		MouseEventCS
}		
}		
if(KeyboardFlag) {		
NumOfKeyboard		vluimsbf5
for(k=0; k< NumOfKeyboard; k++) {		
keyCodeFlag[k]	1	bslbf
if(keyCodeFlag[k]) {		
keyCode	8	uimsbf
}		
event[k]	2	bslbf
}		
}		
if(UserDefinedInputFlag) {		
NumOfUserDefinedInput		vluimsbf5
for(k=0; k<NumOfUserDefinedInput; k++) {		
UserDefinedInput[k]	See ISO 10646	UTF-8
}		
}		
eventID	See ISO 10646	UTF-8
}		

4.4.16.3 Semantics

Name	*Definition*
`VWOEventType`	A type that contains the descriptions of an input event associated to the virtual world object.
`MouseFlag`	This field, which is only present in the binary representation, signals the presence of the `mouse` element. “1” means the element shall be used, and “0” means the element shall not be used.
`KeyboardFlag`	This field, which is only present in the binary representation, signals the presence of the `keyboard` element. “1” means the element shall be used, and “0” means the element shall not be used.
`UserDefinedInputFlag`	This field, which is only present in the binary representation, signals the

<table>
<tr><td></td><td>presence of the UserDefinedInput element. “1” means the element shall be used, and “0” means the element shall not be used.</td></tr>
<tr><td>NumOfMouse</td><td>This field, which is only present in the binary representation, specifies the number of mouse events contained in the VWOEventType.</td></tr>
<tr><td>Mouse</td><td>Describes a mouse event as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the MouseEventCS defined in A.2.1.1.
<table>
<tr><th>Name</th><th>Element

(4 bits)</th><th>Description</th></tr>
<tr><td>click</td><td>0000</td><td>Describes the event of click the left button of a mouse. (click)</td></tr>
<tr><td>doubleclick</td><td>0001</td><td>Describes the event of double-click the left button of a mouse. (doubleclick)</td></tr>
<tr><td>leftBtnDown</td><td>0010</td><td>Describes the event which takes place at the moment of holding down the left button of a mouse. (LeftButtonDown)</td></tr>
<tr><td>leftBtnUp</td><td>0011</td><td>Describes the event which takes place at the moment of releasing the left button of a mouse. (LeftButtonUP)</td></tr>
<tr><td>rightBtnDown</td><td>0100</td><td>Describes the event which takes place at the moment of holding down the left button of a mouse. (RightButtonDown)</td></tr>
<tr><td>rightBtnUp</td><td>0101</td><td>Describes the event which takes place at the moment of releasing the left button of a mouse. (RightButtonUP)</td></tr>
<tr><td>move</td><td>0110</td><td>Describes the event which takes place while changing the mouse position. (Move)</td></tr>
<tr><td></td><td>0111 - 1111</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>NumOfKeyboard</td><td>This field, which is only present in the binary representation, specifies the number of keyboard events contained in the VWOEventType.</td></tr>
<tr><td>keyCodeFlag</td><td>This field, which is only present in the binary representation, signals the presence of the keycode input element. “1” means that the element shall be used, and “0” means that the element shall not be used.</td></tr>
<tr><td>keyCode</td><td>Describes the corresponding key code (0-255) of each key.</td></tr>
<tr><td>event</td><td>Describes the keyboard event (pressed, clicked, or released).
In the binary representation, the keyboard events are presented as follows.
(pressed: 00, clicked: 01, released: 10, and reserved: 11)</td></tr>
<tr><td>NumOfUserDefinedInput</td><td>This field, which is only present in the binary representation, specifies the number of user-defined input events contained in the VWOEventType.</td></tr>
<tr><td>UserDefinedInput</td><td>Describes an input event defined by user.</td></tr>
<tr><td>eventID</td><td>A unique identifier of the event.</td></tr>
</table>

4.4.16.4 Examples

EXAMPLE 1 This example shows the description of an input event with the following semantics. The mouse as an input device produces new input value, “click.” For identifying this input, the value of eventID is “EventID1.”

```
<vwoc:Event eventID="EventID1">
    <vwoc:Mouse>urn:mpeg:mpeg-v:01-VWOC-MouseEventCS-NS:click</vwoc:Mouse>
</vwoc:Event>
```

EXAMPLE 2 This example shows the description of an input event with the following semantics. The Keyboard as an input device produces a new input value which is pressing the key code of “65”. For identifying this input, the value of eventID is “EventID2.”

```
<vwoc:Event eventID="EventID2">
    <vwoc:Keyboard keyCode="65" event="pressed"/>
</vwoc:Event>
```

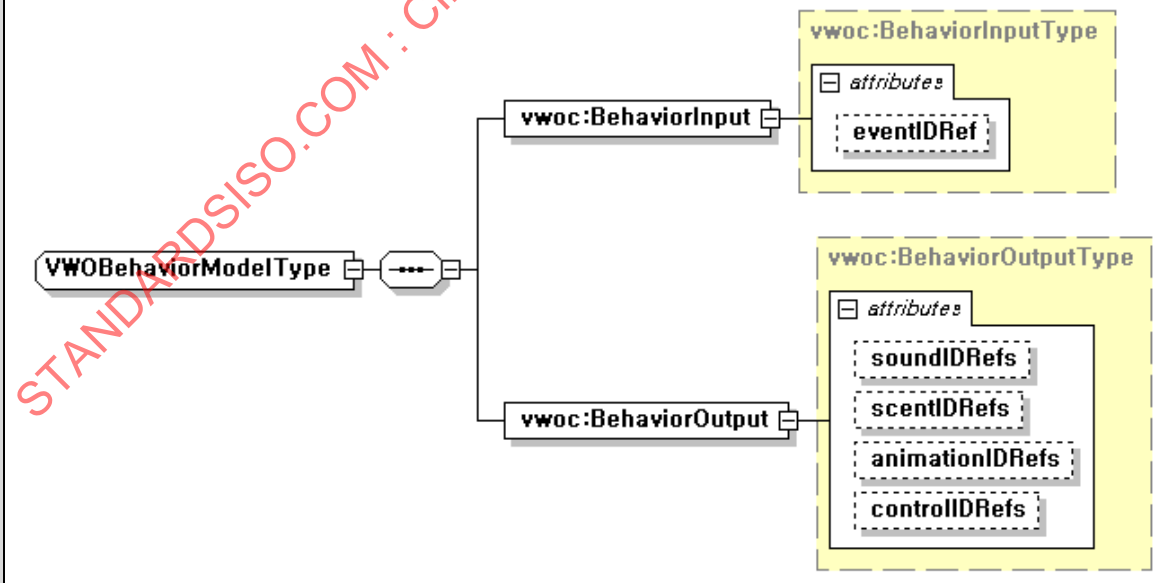
EXAMPLE 3 This example shows the description of an input event with the following semantics. The Keyboard produces a new input event of pressing the two keys “shift” + “a”. One of the keyboard events is the pressing event, “pressed”, of the “shift” key whose code is “16”, and the other one is the pressing event, “pressed”, of the “a” key whose code is “65”. For identifying this input, the value of eventID is “EventID3.”

```
<vwoc:Event eventID="EventID3">
    <vwoc:Keyboard keyCode="16" event="pressed"/>
    <vwoc:Keyboard keyCode="65" event="pressed"/>
</vwoc:Event>
```

4.4.17 VWOBehaviourModelType

4.4.17.1 XML representation syntax

Diagram



Source

```
<complexType name="VWOBehaviorModelType">
    <sequence>
        <element name="BehaviorInput" type="vwoc:BehaviorInputType"/>
        <element name="BehaviorOutput" type="vwoc:BehaviorOutputType"/>
    </sequence>
</complexType>
<complexType name="BehaviorInputType">
    <attribute name="eventIDRef" type="IDREF"/>
</complexType>
```



```
<complexType name="BehaviorOutputType">
    <attribute name="soundIDRefs" type="IDREFS" use="optional"/>
    <attribute name="scentIDRefs" type="IDREFS" use="optional"/>
    <attribute name="animationIDRefs" type="IDREFS" use="optional"/>
    <attribute name="controlIDRefs" type="IDREFS" use="optional"/>
</complexType>
```

4.4.17.2 Binary representation syntax

VWOBehaviorModelType{	***Number of bits***	***Mnemonic***
BehaviorInput		BehaviorInputType
BehaviorOutput		BehaviorOutputType
}		
BehaviorInputType{		
EventIDRefFlag	1	bslbf
if(EventIDRefFlag){		
eventIDRef	See ISO 10646	UTF-8
}		
}		
BehaviorOutputType{		
SoundIDFlag	1	bslbf
ScentIDFlag	1	bslbf
AnimationIDFlag	1	bslbf
ControlIDFlag	1	bslbf
if(SoundIDFlag) {		
SoundIDRefs	See ISO 10646	UTF-8
}		
if(ScentIDFlag) {		
ScentIDRefs	See ISO 10646	UTF-8
}		
if(AnimationIDFlag) {		
AnimationIDRefs	See ISO 10646	UTF-8

}		
if(ControlIDFlag) {		
ControlIDRefs	See ISO 10646	UTF-8
}		
}		

4.4.17.3 Semantics

Name	*Description*
`VWOBehaviorModelType`	A type that describes a container of an input event and the associated output object behaviors.
`BehaviorInput`	An input event to make an object behavior.
`BehaviorInputType`	Refers to an input event ID. *Element* — *Information* eventIDRef — Input event ID EventIDRef Flag — This field, which is only present in the binary representation, signals the presence of the eventIDRef element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`BehaviorOutput`	Object behavior output according to an input event.
`BehaviorOutputType`	Refers to a list of object behavioral outputs. *Element* — *Information* SoundFlag — This field, which is only present in the binary representation, signals the presence of the `sound` element. ”1” means that the element shall be used. ”0” means that the element shall not be used. ScentFlag — This field, which is only present in the binary representation, signals the presence of the `scent` element. ”1” means that the element shall be used. ”0” means that the element shall not be used. AnimationFlag — This field, which is only present in the binary representation, signals the presence of the `animation` element. ”1” means that the element shall be used. ”0” means that the element shall not be used. ControlFlag — This field, which is only present in the binary representation, signals the presence of the `control` element. ”1” means that the element shall be used. ”0” means that the element shall not be used. soundIDRefs — It refers soundIDs to provide sound effects of the object. scentIDRefs — It refers scentIDs to provide scent effects of the object. animationIDRefs — It refers animationIDs to provide animation clips of the object. controlIDRefs — It refers controlIDs to provide controls of the object.

4.4.17.4 Examples

This example shows the description of a VWO behavior model with the following semantics. If eventID =”EventID1” is given as BehaviorInput, then BehaviorOutput shall be executed related to soundID=”SoID5” and animationID=”AniID4”.

```
<vwoc:BehaviorModel>
    <vwoc:BehaviorInput eventIDRef="EventID1"/>
    <vwoc:BehaviorOutput animationIDRefs="AniID4" soundIDRefs="SoID5"/>
```



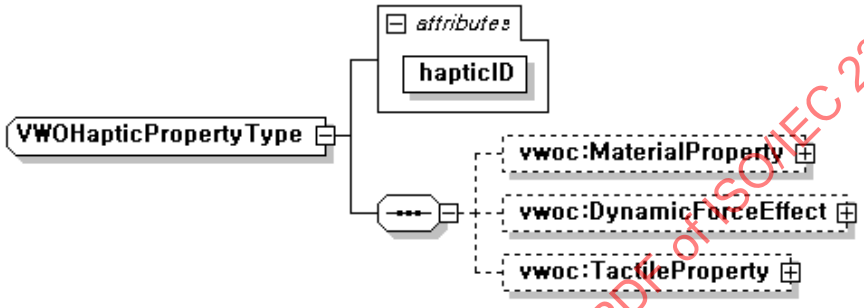
```
</vwoc:BehaviorModel>
```

4.5 Virtual world object common data types

This subclause specifies syntax and semantics of the common datatypes for avatar and virtual object metadata. To be specific, basic data types which are used as basic building blocks, such as haptic properties, animation description, and other simple data types.

4.5.1 VWOHapticPropertyType

4.5.1.1 XML representation syntax

Diagram	
Source	<complexType name="VWOHapticPropertyType"> <sequence> <element name="MaterialProperty" type="vwoc:MaterialPropertyType" minOccurs="0"/> <element name="DynamicForceEffect" type="vwoc:DynamicForceEffectType" minOccurs="0"/> <element name="TactileProperty" type="vwoc:TactileType" minOccurs="0"/> </sequence> <attribute name="hapticID" type="ID" use="required"/> </complexType>

4.5.1.2 Binary representation syntax

VWOHapticPropertyType {	***Number of bits***	***Mnemonic***
MaterialPropertyFlag	1	bslbf
DynamicForceEffectFlag	1	bslbf
TactilePropertyFlag	1	bslbf
if(MaterialPropertyFlag) {		
MaterialProperty		MaterialPropertyType
}		
if(DynamicForceEffectFlag) {		
DynamicForceEffect		DynamicForceEffectType
}		
if(TactilePropertyFlag) {		
TactileProperty		TactilePropertyType

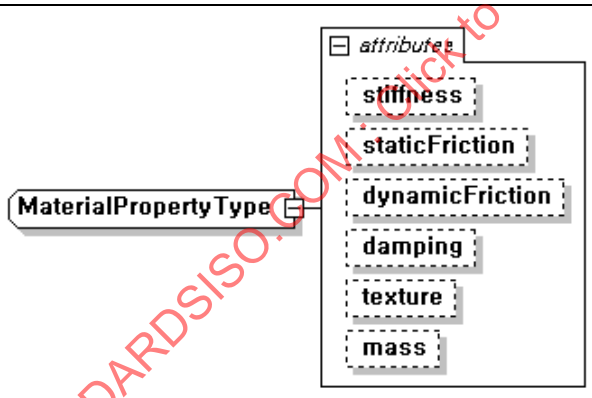
}		
hapticID	See ISO 10646	UTF-8
}		

4.5.1.3 Semantics

Name	*Description*
VWOHapticPropertyType	A type that contains the descriptions of a haptic property associated to the virtual world object.
MaterialPropertyFlag	This field, which is only present in the binary representation, signals the presence of the MaterialProperty element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
DynamicForceEffectFlag	This field, which is only present in the binary representation, signals the presence of the DynamicForceEffect element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
TactilePropertyFlag	This field, which is only present in the binary representation, signals the presence of the TactileProperty element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
MaterialProperty	This type contains parameters characterizing material properties.
DynamicForceEffect	This type contains parameters characterizing force effects.
TactileProperty	This type contains parameters characterizing tactile properties.
hapticID	A unique identifier of the haptic property.

4.5.1.4 MaterialPropertyType

4.5.1.4.1 XML representation syntax

Diagram	
Source	<complexType name="MaterialPropertyType"> <attribute name="stiffness" type="float" use="optional"/> <attribute name="staticFriction" type="float" use="optional"/> <attribute name="dynamicFriction" type="float" use="optional"/> <attribute name="damping" type="float" use="optional"/> <attribute name="texture" type="anyURI" use="optional"/> <attribute name="mass" type="float" use="optional"/> </complexType>

4.5.1.4.2 Binary representation syntax

MaterialPropertyType{	***Number of bits***	***Mnemonic***
StiffnessFlag	1	bslbf
StaticFrictionFlag	1	bslbf
DynamicFrictionFlag	1	bslbf
DampingFlag	1	bslbf
TextureFlag	1	bslbf
MassFlag	1	bslbf
if(StiffnessFlag) {		
stiffness	32	fsbf
}		
if(StaticFrictionFlag) {		
staticFriction	32	fsbf
}		
if(DynamicFrictionFlag) {		
dynamicFriction	32	fsbf
}		
if(DampingFlag) {		
damping	32	fsbf
}		
if(TextureFlag) {		
texture	See ISO 10646	UTF-8
}		
if(MassFlag) {		
mass	32	fsbf
}		
}		

4.5.1.4.3 Semantics

Name	*Description*
MaterialPropertyType	A type that contains the descriptions of a material property associated to the virtual world object.
StiffnessFlag	This field, which is only present in the binary representation, signals the presence of the Stiffness element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
StaticFrictionFlag	This field, which is only present in the binary representation, signals the presence of the StaticFriction element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
DynamicFrictionFlag	This field, which is only present in the binary representation, signals the presence of the DynamicFriction element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
DampingFlag	This field, which is only present in the binary representation, signals the presence of the Damping element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
TextureFlag	This field, which is only present in the binary representation, signals the presence of the Texture element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
MassFlag	This field, which is only present in the binary representation, signals the presence of the Mass element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
stiffness	The stiffness of the virtual world object (in N/mm).
staticFriction	The static friction of the virtual world object.
dynamicFriction	The dynamic friction of the virtual world object.
damping	The damping of the virtual world object.
texture	Contains a link to haptic texture file (e.g. bump image).
mass	The mass of the virtual world object.

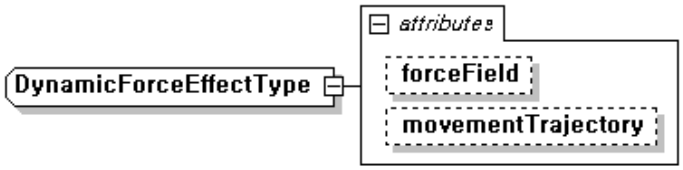
4.5.1.4.4 Examples

This example shows the material properties of a virtual world object which has 0,5 N/mm of stiffness, 0,3 of static coefficient of friction, 0,02 of kinetic coefficient of friction, 0,001 damping coefficient, 0,7 of mass and its surface haptic texture is loaded from the given URL with the id of MID30.

```
<vwoc:HapticProperty hapticID="MID30">
   <vwoc:MaterialProperty stiffness="0.5" staticFriction="0.3"
dynamicFriction="0.02"
damping="0.001" texture="http://haptic.kr/tactile/texture1.bmp" mass="0.7"/>
</vwoc:HapticProperty>
```

4.5.1.5 DynamicForceEffectType

4.5.1.5.1 XML representation syntax

Diagram	
Source	<complexType name="DynamicForceEffectType"> <attribute name="forceField" type="anyURI" use="optional"/> <attribute name="movementTrajectory" type="anyURI" use="optional"/> </complexType>

4.5.1.5.2 Binary representation syntax

DynamicForceEffectType{	*Number of bits*	*Mnemonic*
ForceFieldFlag	1	bslbf
MovementTrajectoryFlag	1	bslbf
if(ForceFieldFlag) {		
forceField	See ISO 10646	UTF-8
}		
if(MovementTrajectoryFlag) {		
movementTrajectory	See ISO 10646	UTF-8
}		
}		

4.5.1.5.3 Semantics

Name	*Description*
`DynamicForceEffectType`	A type that contains the descriptions of a dynamic force effect associated to the virtual world object.
`ForceFieldFlag`	This field, which is only present in the binary representation, signals the presence of the `ForceField` element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
`MovementTrajectoryFlag`	This field, which is only present in the binary representation, signals the presence of the `MovementTrajectory` element. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
`forceField`	Contains link to force field vector file (sum of force field vectors).
`movementTrajectory`	Contains link to force trajectory file (e.g. .dat file including a sequence of motion data).

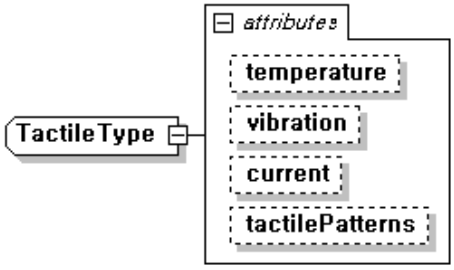
4.5.1.5.4 Examples

This example shows the dynamic force effect of an avatar. The force field characteristic of the avatar with its id of FFID30 is determined by the designed force field file from the URL.

```
<vwoc:HapticProperty hapticID="FFID30">
   <vwoc:DynamicForceEffect forceField="http://haptic.kr/avatar/forcefield.dat"/>
</vwoc:HapticProperty>
```

4.5.1.6 TactileType

4.5.1.6.1 XML representation syntax

Diagram	
Source	<complexType name="TactileType"> <attribute name="temperature" type="float" use="optional"/> <attribute name="vibration" type="float" use="optional"/> <attribute name="current" type="float" use="optional"/> <attribute name="tactilePatterns" type="anyURI" use="optional"/> </complexType>

4.5.1.6.2 Binary representation syntax

TactileType{	***Number of bits***	***Mnemonic***
TemperatureFlag	1	bslbf
VibrationFlag	1	bslbf
CurrentFlag	1	bslbf
TactilePatternsFlag	1	bslbf
if(TemperatureFlag) {		
temperature	32	fsbf
}		
if(VibrationFlag) {		
vibration	32	fsbf
}		
if(CurrentFlag) {		
current	32	fsbf
}		
if(TactilePatternsFlag) {		
tactilePatterns	See ISO 10646	UTF-8
}		
}		

4.5.1.6.3 Semantics

Name	*Description*
`TactileType`	A type that contains the descriptions of a tactile property associated to the virtual world object.
`TemperatureFlag`	This field, which is only present in the binary representation, signals the presence of the `temperature` attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
`VibrationFlag`	This field, which is only present in the binary representation, signals the presence of the `vibration` attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
`CurrentFlag`	This field, which is only present in the binary representation, signals the presence of the electric `current` attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
`TactilePatternsFlag`	This field, which is only present in the binary representation, signals the presence of the `tactilePatterns` attribute. ”1” means the attribute shall be used and ”0” means the attribute shall not be used.
`temperature`	The temperature of the virtual world object (in degree celcius).
`vibration`	The vibration of the virtual world object.
`current`	The electric current of the virtual world object (in mA).
`tactilePatterns`	Contains link to tactile pattern file (e.g. grayscale video, .avi, h.264, or .dat file.)

4.5.1.6.4 Examples

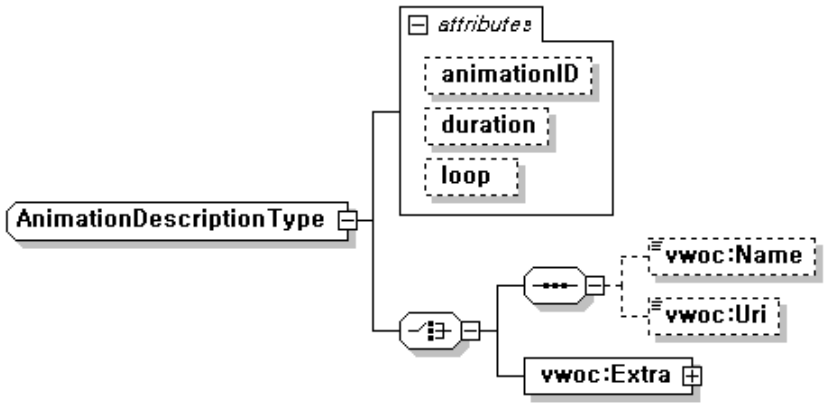
This example shows the tactile properties, with its id of DFEID30, which has 15 degree of temperature and a tactile effect based on the tactile information from the following URL (http://www.haptic.kr/avatar/tactile1.avi).

```
<vwoc:HapticProperty hapticID="DFEID30">
   <vwoc:TactileProperty temperature="15"
tactilePatterns="http://www.haptic.kr/avatar/tactile1.avi"/>
</vwoc:HapticProperty>
```

4.5.2 AnimationDescriptionType

4.5.2.1 XML representation syntax

Diagram



Source	<complexType name="AnimationDescriptionType"> <choice> <sequence> <element name="Name" type="mpeg7:termReferenceType" minOccurs="0"/> <element name="Uri" type="anyURI" minOccurs="0"/> </sequence> <element name="Extra" type="vwoc:ExtraType"/> </choice> <attribute name="animationID" type="ID" use="optional"/> <attribute name="duration" type="unsignedInt" use="optional"/> <attribute name="loop" type="unsignedInt" use="optional"/> </complexType>

4.5.2.2 Binary representation syntax

AnimationDescriptionType{	***Number of bits***	***Mnemonic***
animationIDFlag	1	bslbf
durationFlag	1	bslbf
loopFlag	1	bslbf
SelectAnimationDescriptionType	1	bslbf
if(animationIDFlag){		
animationID	See ISO 10646	UTF-8
}		
if(durationFlag){		
duration	32	uimsbf
}		
if(loopFlag){		
Loop	8	uimsbf
}		
if(SelectAnimationDescriptionType){		
Extra		ExtraType
}		
else {		
NameFlag	1	bslbf
UriFlag	1	bslbf
if(NameFlag){		
TypeOfAnimationCS	8	bslbf

Name	10	Number of bits are defined by the type of AnimationCS
}		
if(UriFlag){		
Uri	See ISO 10646	UTF-8
}		
}		
}		

4.5.2.3 Semantics

Name	*description*
AnimationDescriptionType	A type that contains descriptions and a link to the animation file.
animationIDFlag	This field, which is only present in the binary representation, signals whether animationID attribute is used or not. "1" means that the attribute shall be used, and "0" means that attribute shall not be used.
durationFlag	This field, which is only present in the binary representation, signals whether duration attribute is used or not. "1" means that the attribute shall be used, and "0" means that attribute shall not be used.
loopFlag	This field, which is only present in the binary representation, signals whether loop attribute is used or not. "1" means that the attribute shall be used, and "0" means that attribute shall not be used.
SelectAnimationDescriptionType	This field, which is only present in the binary representation, signals which type of animation description shall be used. "0" means that the proprietary description shall be used, and "1" means that the specified name in the classification schemes and the URI shall be used to describe the animation.
animationID	A unique identifier of the animation.
duration	The length of time that the animation lasts. The default unit is ms.
loop	A playing option to describe the number of repetition. (default value: 1, 0:indefinite repetition, 1:once, 2: twice, …, n: n times)
Extra	Describes an animation in the form of any proprietary but well-formed XML metadata.
NameFlag	This field, which is only present in the binary representation, signals whether the name element is used or not. "1" means that the element shall be used, and "0" means that element shall not be used.
UriFlag	This field, which is only present in the binary representation, signals whether the Uri element is used or not. "1" means that the element shall be used, and "0" means that element shall not be used.
TypeOfAnimationCS	This field, which is only present in the binary representation, describes a class of the animation as one of the classification schemes (CSs). The CSs that may be used for this purpose is defined in A.2.3 and A.2.4.

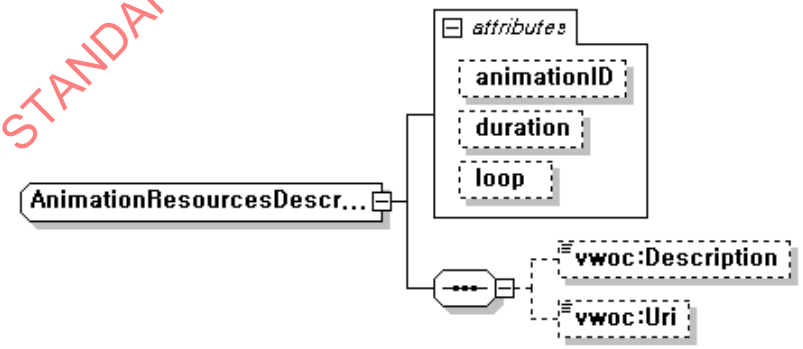
Type of AnimationCS	*Binary representation for sensor type (8 bits)*
IdleAnimationCS	00000000

	GreetingAnimationCS	00000001
	DanceAnimationCS	00000010
	WalkAnimationCS	00000011
	MovesAnimationCS	00000100
	FightingAnimationCS	00000101
	HearingAnimationCS	00000110
	SmokeAnimationCS	00000111
	CongratulationsAnimationCS	00001000
	CommonActionsAnimationCS	00001001
	SpecificActionsAnimationCS	00001010
	FacialExpressionAnimationCS	00001011
	BodyExpressionAnimationCS	00001100
	VODeformationCS	00001101
	VOMotionCS	00001110
	Reserved	00001111-11111111
Name	Describes a type of the animation as a reference to classification schemes (CSs) term. The CSs that may be used for this purpose is defined in A.2.3 and A.2.4.	
Uri	Contains a link to an animation file, usually MP4 file.	

4.5.3 AnimationResourcesDescriptionType

4.5.3.1 XML representation syntax

Diagram



Source

```
<complexType name="AnimationResourcesDescriptionType">
    <sequence>
        <element name="Description" type="string" minOccurs="0"/>
        <element name="Uri" type="anyURI" minOccurs="0"/>
    </sequence>
    <attribute name="animationID" type="ID" use="optional"/>
    <attribute name="duration" type="unsignedInt" use="optional"/>
    <attribute name="loop" type="unsignedInt" use="optional"/>
```

	</complexType>

4.5.3.2 Binary representation syntax

AnimationResourcesDescriptionType{	***Number of bits***	***Mnemonic***
animationIDFlag	1	bslbf
durationFlag	1	bslbf
loopFlag	1	bslbf
DescriptionFlag	1	bslbf
UriFlag	1	bslbf
if(animationIDFlag){		
animationID	See ISO 10646	UTF-8
}		
if(durationFlag){		
duration	8	uimsbf
}		
if(loopFlag){		
loop	8	uimsbf
}		
if(DescriptionFlag){		
Description	See ISO 10646	UTF-8
}		
if(UriFlag){		
Uri	See ISO 10646	UTF-8
}		
}		

4.5.3.3 Semantics

Name	*Description*
`AnimationResourcesDescriptionType`	A type that contains a link to an animation file and its description.
`Description`	Contains the description of the animation resource.

Uri	Contains a link to an animation file, usually MP4 file.
animationID	A unique identifier of the animation.
duration	The length of time that the animation lasts.
loop	A playing option to describe the number of repetition. (default value: 1, 0: indefinite repetition, 1: once, 2: twice, …, n: n times)

4.5.3.4 PointType

4.5.3.4.1 XML representation syntax

Diagram	-
Source	<complexType name="PointType" abstract="true"/> <complexType name="LogicalPointType"> <complexContent> <extension base="vwoc:PointType"> <attribute name="name" type="string" use="optional"/> <attribute name="sensorID" type="anyURI" use="optional"/> </extension> </complexContent> </complexType> <complexType name="Physical3DPointType"> <complexContent> <extension base="vwoc:PointType"> <attribute name="x" type="float" use="required"/> <attribute name="y" type="float" use="required"/> <attribute name="z" type="float" use="required"/> </extension> </complexContent> </complexType>

4.5.3.4.2 Binary representation syntax

PointType{	***Number of bits***	***Mnemonic***
PointTypeSelect	1	bslbf
if (PointTypeSelect) {		
Point		LogicalPointType
}		
else{		
Point		Physical3DPointType
}		
}		
LogicalPointType {		
nameflag	1	bslbf

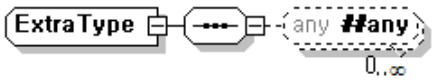
sensorIDflag	1	bslbf
if(nameflag){		
name	See ISO 10646	UTF-8
if(sensorIDflag){		
sensorID	See ISO 10646	UTF-8
}		
}		
Physical3DPointType{		
x	32	fsbf
y	32	fsbf
z	32	fsbf
}		

4.5.3.4.3 Semantics

Name	*Description*
`PointType`	An abstract type providing root for two different point types, which are `LogicalPointType` and `Physical3DPointType` for specifying a feature point for face feature control.
`PointTypeSelect`	This field, which is only present in the binary representation, signals whether the type of point is logical point type or the type of point is a physical 3d positional point.”1” means that the logical point type shall be used, and ”0” means that the physical 3D point type shall be used.
`LogicalPointType`	A type providing the name of the feature point.
`nameflag`	This field, which is only present in the binary representation, signals whether `name` is used or not. ”1” means that the name shall be used, and ”0” means that name shall not be used.
`sensorIDflag`	This field, which is only present in the binary representation, signals whether `sensorID` is used or not. ”1” means that the sensorID shall be used, and ”0” means that the sensorID shall not be used.
`name`	The name of the feature point.
`sensorID`	The sensor ID corresponding to the feature point.
`Physical3DPointType`	A type providing a three dimensional point vector value.
`x`	The point value on x-axis in 3 dimensional space.
`y`	The point value on y-axis in 3 dimensional space.
`z`	The point value on z-axis in 3 dimensional space.

4.5.3.5 ExtraType

4.5.3.5.1 XML representation syntax

Diagram	
Source	<complexType name="ExtraType"> <annotation> <appinfo>enable-xmlns</appinfo> </annotation> <sequence> <any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType>

4.5.3.5.2 Binary representation syntax

ExtraType {	***Number of bits***	***Mnemonic***
XMLDataFlag	1	bslbf
if (XMLDataFlag) {		
NumXMLData		vluimsbf5
for(k=0; k<NumXMLData; k++) {		
XMLLength		vluimsbf5
XMLData	See ISO 10646	UTF-8
}		
}		
}		

4.5.3.5.3 Semantics

Name	*Description*
`ExtraType`	A type that can contain any well-formed XML data.
`XMLDataFlag`	This field, which is only present in the binary representation, signals whether XML data is used or not. ”1” means that the XML data shall be used, and ”0” means that XML data shall not be used.
`NumXMLData`	This field, which is only present in the binary representation, specifies the number of XML data contained in the `ExtraType` element.
`XMLLength`	This field, which is only present in the binary representation, specifies the number of characters in terms of UTF-8 in each XML data.
`XMLData`	This field contains any well-formed XML data. In the binary representation, the field is encoded by UTF-8.

Note: Element defined as type <`ExtraType`> allows extending the MPEG-V schema with proprietary but well defined or at least well formatted data.

4.5.4 Common simple data types

4.5.4.1 IndicateOfLHType

4.5.4.1.1 XML representation syntax

Source	<simpleType name="indicateOfLHType"> <restriction base="string"> <enumeration value="low"/> <enumeration value="high"/> </restriction> </simpleType>

4.5.4.1.2 Binary representation syntax

	Number of bits	***Mnemonic***
indicateOfLHType	1	bslbf

4.5.4.1.3 Semantics

Name	*Description*
`indicateOfLHType`	A type of which the value is either low or high. The binary representation of the type is defined as follows. (0: low, 1: high)

4.5.4.2 IndicateOfLMHType

4.5.4.2.1 XML representation syntax

Source	<simpleType name="indicateOfLMHType"> <restriction base="string"> <enumeration value="low"/> <enumeration value="medium"/> <enumeration value="high"/> </restriction> </simpleType>

4.5.4.2.2 Binary representation syntax

	Number of bits	***Mnemonic***
indicateOfLMHType	2	bslbf

4.5.4.2.3 Semantics

Name	*Description*
`indicateOfLMHType`	A type of which the value is among low, medium or high. The binary representation of the type is defined as follows. (0: low, 1: medium, 2: high, 3: reserved)

4.5.4.3 IndicateOfSMBType

4.5.4.3.1 XML representation syntax

Source	<simpleType name="indicateOfSMBType"> <restriction base="string"> <enumeration value="small"/> <enumeration value="medium"/> <enumeration value="big"/> </restriction> </simpleType>

4.5.4.3.2 Binary representation syntax

	Number of bits	*Mnemonic*
indicateOfSMBType	2	bslbf

4.5.4.3.3 Semantics

Name	*Description*
`indicateOfSMBType`	A type of which the value is among small, medium or big. The binary representation of the type is defined as follows. (0: small, 1: medium, 2: big, 3: reserved)

4.5.4.4 IndicateOfSMLType

4.5.4.4.1 XML representation syntax

Source	<simpleType name="indicateOfSMLType"> <restriction base="string"> <enumeration value="short"/> <enumeration value="medium"/> <enumeration value="long"/> </restriction> </simpleType>

4.5.4.4.2 Binary representation syntax

	Number of bits	*Mnemonic*
indicateOfSMLType	2	bslbf

4.5.4.4.3 Semantics

Name	*Description*
`indicateOfSMLType`	A type of which the value is among short, medium or long. The binary representation of the type is defined as follows. (0: short, 1: medium, 2: long, 3: reserved)

4.5.4.5 IndicateOfDMUType

4.5.4.5.1 XML representation syntax

Source	<simpleType name="indicateOfDMUType"> <restriction base="string"> <enumeration value="down"/>


```
            <enumeration value="medium"/>
            <enumeration value="up"/>
        </restriction>
    </simpleType>
```

4.5.4.5.2 Binary representation syntax

	Number of bits	*Mnemonic*
indicateOfDMUType	2	bslbf

4.5.4.5.3 Semantics

Name	*Description*
`indicateOfDMUType`	A type of which the value is among down, medium or up. The binary representation of the type is defined as follows. (0: down, 1: medium, 2: up, 3: reserved)

4.5.4.6 IndicateOfDUType

4.5.4.6.1 XML representation syntax

Source

```
<simpleType name="indicateOfDUType">
    <restriction base="string">
        <enumeration value="down"/>
        <enumeration value="up"/>
    </restriction>
</simpleType>
```

4.5.4.6.2 Binary representation syntax

	Number of bits	*Mnemonic*
indicateOfDUType	1	bslbf

4.5.4.6.3 Semantics

Name	*Description*
`indicateOfDUType`	A type of which the value is either down or up. The binary representation of the type is defined as follows. (0: down, 1: up)

4.5.4.7 IndicateOfPMNType

4.5.4.7.1 XML representation syntax

Source

```
<simpleType name="indicateOfPMNType">
    <restriction base="string">
        <enumeration value="pointed"/>
        <enumeration value="middle"/>
        <enumeration value="notpointed"/>
    </restriction>
</simpleType>
```

4.5.4.7.2 Binary representation syntax

	Number of bits	*Mnemonic*
indicateOfPMNType	2	bslbf

4.5.4.7.3 Semantics

Name	*Description*
`indicateOfPMNType`	A type of which the value is among pointed, middle or not pointed. The binary representation of the type is defined as follows. (0: short, 1: medium, 2: long, 3: reserved)

4.5.4.8 IndicateOfRCType

4.5.4.8.1 XML representation syntax

Source	<simpleType name="indicateOfRCType"> <restriction base="string"> <enumeration value="round"/> <enumeration value="cleft"/> </restriction> </simpleType>

4.5.4.8.2 Binary representation syntax

	Number of bits	*Mnemonic*
indicateOfRCType	1	bslbf

4.5.4.8.3 Semantics

Name	*Description*
`indicateOfRCType`	A type of which the value is either round or cleft. The binary representation of the type is defined as follows. (0: round, 1: cleft)

4.5.4.9 IndicateOfLRType

4.5.4.9.1 XML representation syntax

Source	<simpleType name="indicateOfLRType"> <restriction base="string"> <enumeration value="left"/> <enumeration value="right"/> </restriction> </simpleType>

4.5.4.9.2 Binary representation syntax

	Number of bits	*Mnemonic*
indicateOfLRType	1	bslbf

4.5.4.9.3 Semantics

Name	*Description*
`indicateOfLRType`	A type of which the value is either left or right. The binary representation of the type is defined as follows. (0: left, 1: right)

4.5.4.10 IndicateOfLMRType

4.5.4.10.1 XML representation syntax

Source	<simpleType name="indicateOfLMRType"> <restriction base="string"> <enumeration value="left"/> <enumeration value="middle"/> <enumeration value="right"/> </restriction> </simpleType>

4.5.4.10.2 Binary representation syntax

	Number of bits	***Mnemonic***
indicateOfLMRType	2	bslbf

4.5.4.10.3 Semantics

Name	*Description*
`indicateOfLMRTyp e`	A type of which the value is amoung left, middle or right. The binary representation of the type is defined as follows. (0: left, 1: middle, 2: right, 3: reserved)

4.5.4.11 measureUnitLMHType

4.5.4.11.1 XML representation syntax

Source	<simpleType name="measureUnitLMHType"> <union memberTypes="vwoc:indicateOfLMHType float"/> </simpleType>

4.5.4.11.2 Binary representation syntax

measureUnitLMHType{	***Number of bits***	***Mnemonic***
selectType	1	bslbf
if(selectType == 0){		
indicationOfLMH		indicateOfLMHType
}else{		

measure	32	fsbf
}		
}		

4.5.4.11.3 Semantics

Name	*Description*
`measureUnitLMHType`	A type which may be either `indicateOfLMHType` or float.
`selectType`	This field, which is only present in the binary representation, signals whether a floating point value is used or the `indicateOfLMHType` is used. ”1” means that the `indicateOfLMHType` shall be used, and ”0” means that a floating point value shall not be used.
`measure`	This field, which is only present in the binary representation, the value of which is a floating point value.

4.5.4.12 measureUnitSMBType

4.5.4.12.1 XML representation syntax

Source	<simpleType name="measureUnitSMBType"> <union memberTypes="vwoc:indicateOfSMBType float"/> </simpleType>

4.5.4.12.2 Binary representation syntax

measureUnitSMBType{	***Number of bits***	***Mnemonic***
selectType	1	bslbf
if(selectType == 0){		
indicateOfSMB		indicateOfSMBType
}else{		
measure	32	fsbf
}		
}		

4.5.4.12.3 Semantics

Name	*Description*
`measureUnitSMBType`	A type which may be either `indicateOfSMBType` or float.
`selectType`	This field, which is only present in the binary representation, signals whether a floating point value is used or the `indicateOfSMBType` is used. ”1” means that the `indicateOfSMBType` shall be used, and ”0” means that a floating point value shall not be used.
`measure`	This field, which is only present in the binary representation, the value

	of which is a floating point value.

4.5.4.13 levelOf5Type

4.5.4.13.1 XML representation syntax

Source	<simpleType name="levelOf5Type"> <restriction base="integer"> <minInclusive value="1"/> <maxInclusive value="5"/> </restriction> </simpleType>

4.5.4.13.2 Binary representation syntax

	Number of bits	*Mnemonic*
levelOf5Type	3	uimsbf

4.5.4.13.3 Semantics

Name	*Description*
`levelOf5Type`	A type of which the integer value is from one to five. The binary representation of the type is defined as follows. (0:0, 1:1, 2:2, 3:3, 4:4, 5:5, 6-8:reserved)

4.5.4.14 angleType

4.5.4.14.1 XML representation syntax

Source	<simpleType name="angleType"> <restriction base="float"> <minInclusive value="0"/> <maxInclusive value="360"/> </restriction> </simpleType>

4.5.4.14.2 Binary representation syntax

	Number of bits	*Mnemonic*
angleType	32	fsbf

4.5.4.14.3 Semantics

Name	*Description*
`angleType`	A type of which the floating point value is from 0 degree to 360 degree.

4.5.4.15 percentageType

4.5.4.15.1 XML representation syntax

Source	<simpleType name="percentageType"> <restriction base="float"> <minInclusive value="0"/> <maxInclusive value="100"/> </restriction> </simpleType>

4.5.4.15.2 Binary representation syntax

	Number of bits	Mnemonic
percentageType	32	fsbf

4.5.4.15.3 Semantics

Name	*Description*
percentageType	A type of which the floating point value is from 0 percent to 100 percent.

4.5.4.16 unlimitedPertageType

4.5.4.16.1 XML representation syntax

Source	<simpleType name="unlimitedPercentageType"> <restriction base="float"> <minInclusive value="0"/> </restriction> </simpleType>

4.5.4.16.2 Binary representation syntax

	Number of bits	***Mnemonic***
unlimitedPercentageType	32	fsbf

4.5.4.16.3 Semantics

Name	*Description*
unlimitedPercentageType	A type of which the floating point value is from 0 percent.

5 Avatar metadata

5.1 General

Avatar metadata as a (visual) representation of the user inside the environment serves the following purposes:

— make visible the presence of a real user into the VE;

— characterize the user within the VE;

— interact with the VE.

The "Avatar" element is composed of following type of data with the extension of the base type of avatar.

— **Appearance**: contains the high-level description of the appearance and may refer a media containing the exact geometry and texture.

— **Animation**: contains the description of a set of animation sequences that the avatar is able to perform and may refer to several media containing the exact (geometric transformations) animation parameters.

— **CommunicationSkills**: contains a set of descriptors providing information on the different modalities an avatar is able to communicate.

— **Personality**: contains a set of descriptors defining the personality of the avatar.

— **ControlFeatures**: contains a set of descriptors defining possible place-holders for sensors on body skeleton and face feature points.

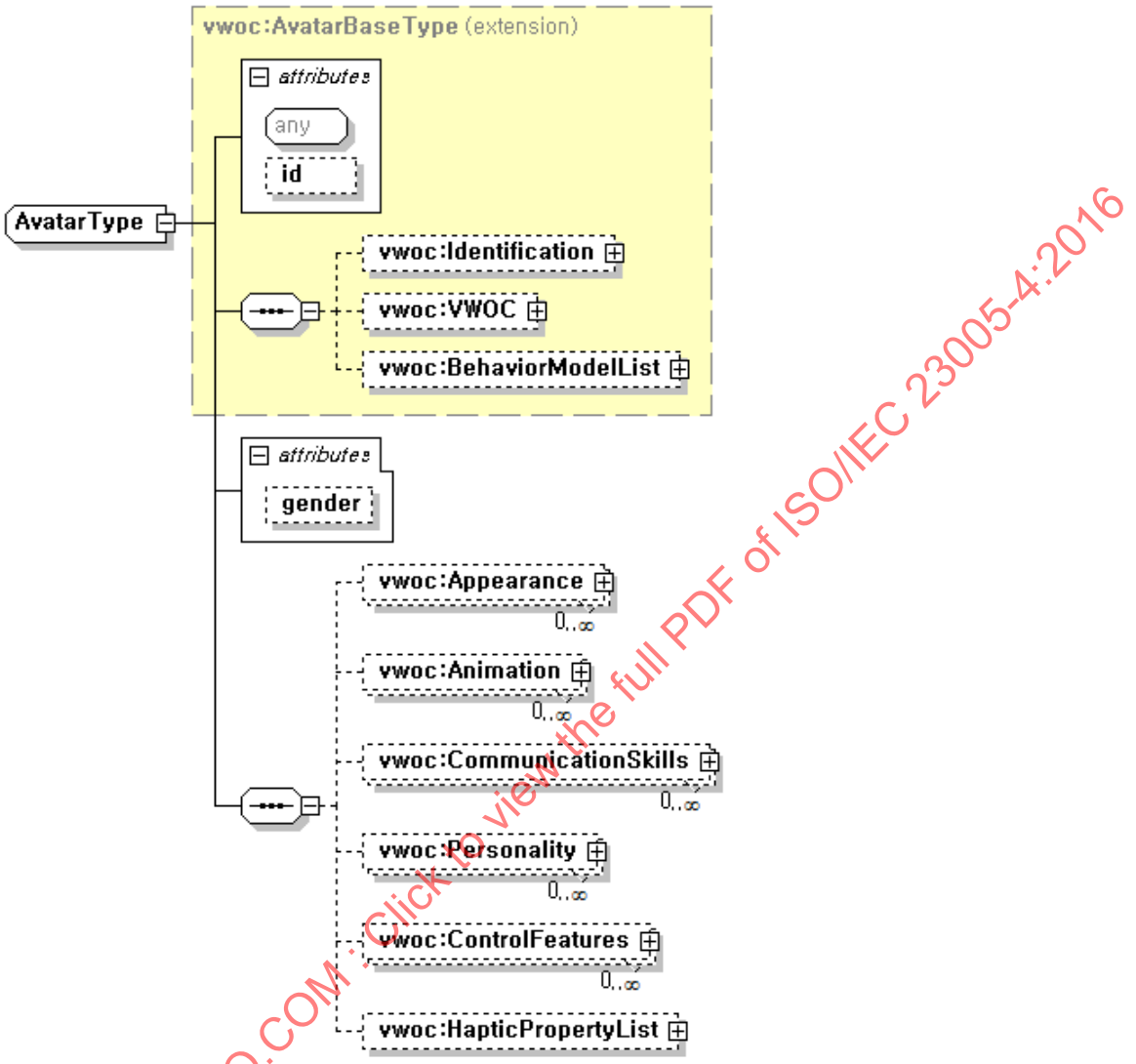
— **HapticPropertyList**: contains a list of high level descriptors of the haptic properties.

— **gender**: describes the gender of the avatar.

5.2 AvatarType

5.2.1 XML representation syntax

Diagram



Source

```
<complexType name="AvatarType">
    <complexContent>
        <extension base="vwoc:AvatarBaseType">
            <sequence>
                <element name="Appearance" type="vwoc:AvatarAppearanceType" minOccurs="0"
maxOccurs="unbounded"/>
                <element name="Animation" type="vwoc:AvatarAnimationType" minOccurs="0"
maxOccurs="unbounded"/>
                <element name="CommunicationSkills" type="vwoc:AvatarCommunicationSkillsType"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="Personality" type="vwoc:AvatarPersonalityType" minOccurs="0"
maxOccurs="unbounded"/>
                <element name="ControlFeatures" type="vwoc:AvatarControlFeaturesType"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="HapticPropertyList" type="vwoc:VWOHapticPropertyListType"
minOccurs="0"/>
            </sequence>
            <attribute name="gender" type="string" use="optional"/>
        </extension>
    </complexContent>
</complexType>
```

5.2.2 Binary representation syntax

AvatarType{	*Number of bits*	*Mnemonic*
AvatarBase		AvatarBaseType
AppereanceFlag	1	bslbf
AnimationFlag	1	bslbf
CommunicationSkillsFlag	1	bslbf
PersonalityFlag	1	bslbf
ControlFeaturesFlag	1	bslbf
HapticPropertyListFlag	1	bslbf
genderFlag	1	bslbf
if(AppereanceFlag){		
LoopAvatarAppereance		vluimsbf5
for(k=0; k< LoopAvatarAppereance; k++){		
Appereance[k]		AvatarAppereanceType
}		
}		
if(AnimationFlag){		
LoopAvatarAnimation		vluimsbf5
for(k=0; k<LoopAvatarAnimation; k++){		
Animation[k]		AvatarAnimationType
}		
}		
if(CommunicationSkillsFlag){		
LoopAvatarCommunicationSkills		vluimsbf5
for(k=0;k<LoopAvatarCommunicationSkills; k++){		

CommunicationSkills[k]		AvatarCommunicationSkillsType
}		
}		
if(PersonalityFlag){		
LoopAvatarPersonality		vluimsbf5
for(k=0;k<LoopAvatarPersonality; k++){		
Personality[k]		AvatarPersonalityType
}		
}		
if(ControlFeaturesFlag){		
LoopAvatarControlFeatures		vluimsbf5
for(k=0;k<LoopAvatarControlFeatures; k++){		
ControlFeatures[k]		AvatarControlFeaturesType
}		
}		
if(HapticPropertyListFlag){		
HapticPropertyList		VWOHapticPropertyListType
}		
if(genderFlag){		
gender	See ISO 10646	UTF-8
}		
}		

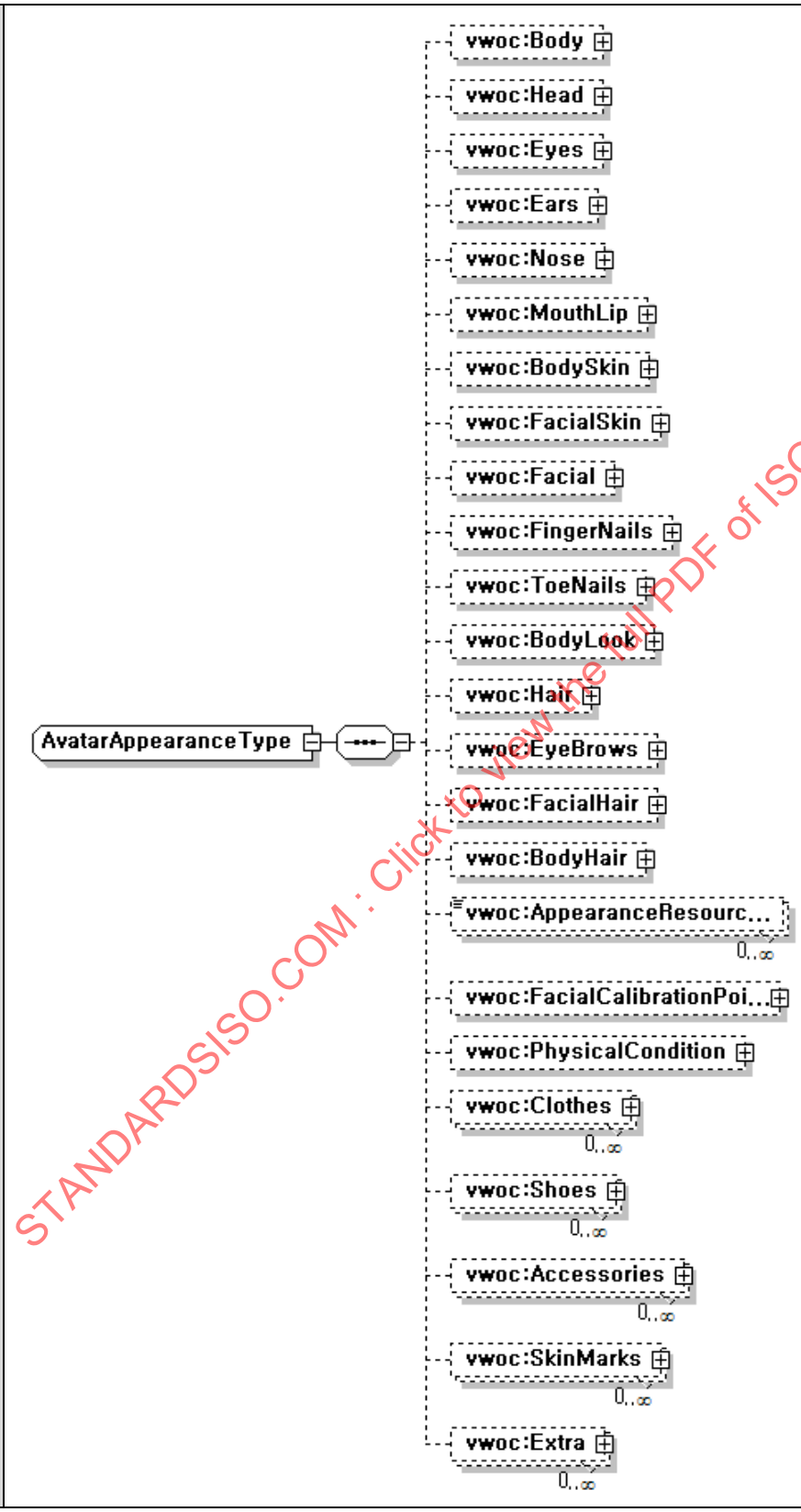
5.2.3 Semantics

Name	*Description*
AvatarType	A type that represents the user inside the virtual world environment.
AvatarBase	Contains the base type defined by AvatarBaseType.
AppearanceFlag	This field, which is only present in the binary representation, signals the presence of the Appearance elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.
AnimationFlag	This field, which is only present in the binary representation, signals the presence of the Animation elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.
CommunicationSkillsFlag	This field, which is only present in the binary representation, signals the presence of the CommunicationSkills elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.
PersonalityFlag	This field, which is only present in the binary representation, signals the presence of the Personality elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.
ControlFeaturesFlag	This field, which is only present in the binary representation, signals the presence of the ControlFeatures elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.
HapticPropertyListFlag	This field, which is only present in the binary representation, signals the presence of the HapticPropertyList elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.
genderFlag	This field, which is only present in the binary representation, signals the presence of the gender attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
LoopAvatarAppearance	This field, which is only present in the binary representation, specifies the number of appearance information contained in the avatar characteristics.
Appearance	Contains the high level description of the appearance of an avatar.
LoopAvatarAnimation	This field, which is only present in the binary representation, specifies the number of animation information contained in the avatar characteristics.
Animation	Contains the description of a set of animation sequences that the avatar is able to perform.
LoopAvatarCommunicationSkills	This field, which is only present in the binary representation, specifies the number of communication skills information contained in the avatar characteristics.
CommunicationSkills	Contains a set of descriptors providing information on the different modalities an avatar is able to communicate.
LoopAvatarPersonality	This field, which is only present in the binary representation, specifies the number of personality information contained in the avatar characteristics.
Personality	Contains a set of descriptors defining the personality of the avatar.
LoopAvatarControlFeatures	This field, which is only present in the binary representation, specifies the number of feature control information contained in the avatar characteristics.
ControlFeatures	Contains a set of descriptors defining possible place-holders for sensors on body skeleton and face feature points.
HapticPropertyListFlag	This field, which is only present in the binary representation, signals the presence of the HapticPropertyList. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HapticPropertyList	Contains a list of high level descriptors of the haptic properties.
genderFlag	This field, which is only present in the binary representation, signals the presence of the gender attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
gender	Describes the gender of the avatar.

5.2.4 AvatarAppearanceType

5.2.4.1 XML representation syntax

Diagram



Source

```
<complexType name="AvatarAppearanceType">
    <sequence>
        <element name="Body" type="vwoc:BodyType" minOccurs="0"/>
        <element name="Head" type="vwoc:HeadType" minOccurs="0"/>
        <element name="Eyes" type="vwoc:EyesType" minOccurs="0"/>
        <element name="Ears" type="vwoc:EarsType" minOccurs="0"/>
        <element name="Nose" type="vwoc:NoseType" minOccurs="0"/>
        <element name="MouthLip" type="vwoc:MouthLipType" minOccurs="0"/>
        <element name="BodySkin" type="vwoc:SkinType" minOccurs="0"/>
        <element name="FacialSkin" type="vwoc:SkinType" minOccurs="0"/>
        <element name="Facial" type="vwoc:FacialType" minOccurs="0"/>
        <element name="FingerNails" type="vwoc:NailType" minOccurs="0"/>
        <element name="ToeNails" type="vwoc:NailType" minOccurs="0"/>
        <element name="BodyLook" type="vwoc:BodyLookType" minOccurs="0"/>
        <element name="Hair" type="vwoc:HairType" minOccurs="0"/>
        <element name="EyeBrows" type="vwoc:EyebrowsType" minOccurs="0"/>
        <element name="FacialHair" type="vwoc:FacialHairType" minOccurs="0"/>
        <element name="BodyHair" type="vwoc:BodyHairType" minOccurs="0"/>
        <element name="AppearanceResources" type="anyURI" minOccurs="0" maxOccurs="unbounded"/>
        <element name="FacialCalibrationPoints" type="vwoc:FacialCalibrationPointsType" minOccurs="0"/>
        <element name="PhysicalCondition" type="vwoc:PhysicalConditionType" minOccurs="0"/>
        <element name="Clothes" type="vwoc:VirtualObjectType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Shoes" type="vwoc:VirtualObjectType" minOccurs="0" maxOccurs="unbounded"/>
        <element name="Accessories" type="vwoc:VirtualObjectType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="SkinMarks " type="vwoc:VirtualObjectType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>

<complexType name="BodyType">
    <sequence>
        <element name="BodyHeight" type="float" minOccurs="0"/>
        <element name="BodyThickness" type="float" minOccurs="0"/>
        <element name="BodyFat" type="vwoc:measureUnitLMHType" minOccurs="0"/>
        <element name="TorsoMuscles" type="vwoc:measureUnitLMHType" minOccurs="0"/>
        <element name="NeckThikness" type="float" minOccurs="0"/>
        <element name="NeckLength" type="float" minOccurs="0"/>
        <element name="Shoulders" type="float" minOccurs="0"/>
        <element name="Pectorials" type="float" minOccurs="0"/>
        <element name="ArmLength" type="float" minOccurs="0"/>
        <element name="HeadSize" type="float" minOccurs="0"/>
        <element name="TorsoLength" type="float" minOccurs="0"/>
        <element name="LoveHandles" type="float" minOccurs="0"/>
        <element name="BellySize" type="float" minOccurs="0"/>
        <element name="LegMuscles" type="float" minOccurs="0"/>
        <element name="LegLength" type="float" minOccurs="0"/>
        <element name="HipWidth" type="float" minOccurs="0"/>
        <element name="HipLength" type="float" minOccurs="0"/>
        <element name="ButtSize" type="float" minOccurs="0"/>
        <element name="Package" type="vwoc:indicateOfSMBType" minOccurs="0"/>
        <element name="SaddleBags" type="vwoc:indicateOfSMBType" minOccurs="0"/>
        <element name="KneeAngle" type="vwoc:angleType" minOccurs="0"/>
        <element name="FootSize" type="float" minOccurs="0"/>
        <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="hapticIDRef" type="IDREF" use="optional"/>
</complexType>

<complexType name="HeadType">
    <sequence>
        <element name="HeadSize" type="vwoc:measureUnitSMBType" minOccurs="0"/>
        <element name="HeadStretch" type="vwoc:unlimitedPercentageType" minOccurs="0"/>
        <element name="HeadShape" minOccurs="0">
            <simpleType>
                <restriction base="string">
```

```
                        <enumeration value="square"/>
                        <enumeration value="round"/>
                        <enumeration value="oval"/>
                        <enumeration value="long"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="EggHead" type="boolean" minOccurs="0"/>
            <element name="HeadLength" type="float" minOccurs="0"/>
            <element name="FaceShear" type="float" minOccurs="0"/>
            <element name="ForeheadSize" type="float" minOccurs="0"/>
            <element name="ForeheadAngle" type="vwoc:angleType" minOccurs="0"/>
            <element name="BrowSize" type="float" minOccurs="0"/>
            <element name="FaceSkin" minOccurs="0">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="dry"/>
                        <enumeration value="normal"/>
                        <enumeration value="greassy"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="Cheeks" type="vwoc:measureUnitSMBType" minOccurs="0"/>
            <element name="CheeksDepth" type="float" minOccurs="0"/>
            <element name="CheeksShape" minOccurs="0">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="chubby"/>
                        <enumeration value="high"/>
                        <enumeration value="bone"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="UpperCheeks" type="vwoc:measureUnitSMBType" minOccurs="0"/>
            <element name="LowerCheeks" type="vwoc:measureUnitSMBType" minOccurs="0"/>
            <element name="CheekBones" type="vwoc:indicateOfDMUType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
    </complexType>

    <complexType name="EyesType">
        <sequence>
            <element name="EyeSize" type="float" minOccurs="0"/>
            <element name="EyeOpening" type="vwoc:unlimitedPercentageType" minOccurs="0"/>
            <element name="EyeSpacing" type="float" minOccurs="0"/>
            <element name="OuterEyeCorner" type="vwoc:indicateOfDMUType" minOccurs="0"/>
            <element name="InnerEyeCorner" type="vwoc:indicateOfDMUType" minOccurs="0"/>
            <element name="EyeDepth" type="float" minOccurs="0"/>
            <element name="UpperEyelidFold" type="float" minOccurs="0"/>
            <element name="EyeBags" type="float" minOccurs="0"/>
            <element name="PuffyEyeLids" type="vwoc:indicateOfSMBType" minOccurs="0"/>
            <element name="EyelashLength" type="float" minOccurs="0"/>
            <element name="EyePop" type="float" minOccurs="0"/>
            <element name="EyeColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="EyeLightness" type="vwoc:percentageType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
    </complexType>

    <complexType name="EarsType">
        <sequence>
            <element name="EarSize" type="float" minOccurs="0"/>
            <element name="EarPosition" type="vwoc:indicateOfDMUType" minOccurs="0"/>
            <element name="EarAngle" minOccurs="0">
                <simpleType>
                    <restriction base="vwoc:angleType">
                        <maxInclusive value="180"/>
```

```
                    </restriction>
                </simpleType>
            </element>
            <element name="AttachedEarlobes" type="float" minOccurs="0"/>
            <element name="EarTips" type="vwoc:indicateOfPMNType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
</complexType>

<complexType name="NoseType">
        <sequence>
            <element name="NoseSize" type="float" minOccurs="0"/>
            <element name="NoseWidth" type="float" minOccurs="0"/>
            <element name="NostrillWidth" type="float" minOccurs="0"/>
            <element name="NostrillDivision" type="float" minOccurs="0"/>
            <element name="NoseThickness" type="float" minOccurs="0"/>
            <element name="UpperBridge" type="float" minOccurs="0"/>
            <element name="LowerBridge" type="float" minOccurs="0"/>
            <element name="BridgeWidth" type="float" minOccurs="0"/>
            <element name="NoseTipAngle" type="vwoc:indicateOfDUType" minOccurs="0"/>
            <element name="NoseTipShape" minOccurs="0">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="pointy"/>
                        <enumeration value="bulbous"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="CrookedNose" type="vwoc:indicateOfLRType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
</complexType>

<complexType name="MouthLipType">
        <sequence>
            <element name="LipWidth" type="float" minOccurs="0"/>
            <element name="LipFullness" type="float" minOccurs="0"/>
            <element name="LipThickness" type="float" minOccurs="0"/>
            <element name="LipRatio" type="float" minOccurs="0"/>
            <element name="MouthSize" type="float" minOccurs="0"/>
            <element name="MouthPosition" type="float" minOccurs="0"/>
            <element name="MouthCorner" type="vwoc:indicateOfDMUType" minOccurs="0"/>
            <element name="LipCleftDepth" type="float" minOccurs="0"/>
            <element name="LipCleft" type="float" minOccurs="0"/>
            <element name="ShiftMouth" type="vwoc:indicateOfLMRType" minOccurs="0"/>
            <element name="ChinAngle" minOccurs="0">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="inner"/>
                        <enumeration value="outer"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="JawShape" type="vwoc:indicateOfPMNType" minOccurs="0"/>
            <element name="ChinDepth" type="float" minOccurs="0"/>
            <element name="JawAngle" type="float" minOccurs="0"/>
            <element name="JawJut" minOccurs="0">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="inside"/>
                        <enumeration value="outside"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="Jowls" type="float" minOccurs="0"/>
            <element name="ChinCleft" type="vwoc:indicateOfRCType" minOccurs="0"/>
```

```
            <element name="UpperChinCleft" type="vwoc:indicateOfRCType" minOccurs="0"/>
            <element name="ChinNeck" type="float" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
    </complexType>

    <complexType name="SkinType">
        <sequence>
            <element name="SkinPigment" type="mpegvct:colorType" minOccurs="0"/>
            <element name="SkinRuddiness" type="vwoc:percentageType" minOccurs="0"/>
            <element name="SkinRainbowColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
    </complexType>

    <complexType name="FacialType">
        <sequence>
            <element name="FacialDefinition" type="vwoc:levelOf5Type" minOccurs="0"/>
            <element name="Freckles" type="vwoc:levelOf5Type" minOccurs="0"/>
            <element name="Wrinkles" type="boolean" minOccurs="0"/>
            <element name="RosyComplexion" type="boolean" minOccurs="0"/>
            <element name="LipPinkness" type="vwoc:levelOf5Type" minOccurs="0"/>
            <element name="Lipstick" type="boolean" minOccurs="0"/>
            <element name="LipstickColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="LipGloss" type="vwoc:levelOf5Type" minOccurs="0"/>
            <element name="Blush" type="boolean" minOccurs="0"/>
            <element name="BlushColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="BlushOpacity" type="vwoc:percentageType" minOccurs="0"/>
            <element name="InnerShadow" type="boolean" minOccurs="0"/>
            <element name="InnerShadowColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="InnerShadowOpacity" type="vwoc:percentageType" minOccurs="0"/>
            <element name="OuterShadow" type="boolean" minOccurs="0"/>
            <element name="OuterShadowOpacity" type="vwoc:percentageType" minOccurs="0"/>
            <element name="EyeLiner" type="boolean" minOccurs="0"/>
            <element name="EyeLinerColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
    </complexType>

    <complexType name="NailType">
        <sequence>
            <element name="NailPolish" type="boolean" minOccurs="0"/>
            <element name="NailPolishColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
    </complexType>

    <complexType name="BodyLookType">
        <sequence>
            <element name="BodyDefinition" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="BodyFreckles" type="vwoc:levelOf5Type" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <complexType name="HairType">
        <sequence>
            <element name="HairSize" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="HairStyle" type="mpeg7:termReferenceType" minOccurs="0"/>
            <element name="HairColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="WhiteHair" type="vwoc:percentageType" minOccurs="0"/>
            <element name="RainbowColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="BlondeHair" type="vwoc:percentageType" minOccurs="0"/>
            <element name="RedHair" type="vwoc:percentageType" minOccurs="0"/>
            <element name="HairVolume" type="vwoc:indicateOfSMBType" minOccurs="0"/>
```

```
            <element name="HairFront" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="HairSides" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="HairBack" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="BigHairFront" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="BigHairTop" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="BigHairBack" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="FrontFrindge" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="SideFrindge" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="BackFrindge" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="FullHairSides" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="HairSweep" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="ShearFront" type="vwoc:indicateOfLMRType" minOccurs="0"/>
            <element name="ShearBack" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="TuperFront" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="TuperBack" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="RumpledHair" minOccurs="0">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="low"/>
                        <enumeration value="moderate"/>
                        <enumeration value="high"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="PigTails" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="PonyTail" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="SprikedHair" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="HairTilt" type="float" minOccurs="0"/>
            <element name="HairMiddlePart" type="vwoc:indicateOfLHType" minOccurs="0"/>
            <element name="HairRightPart" type="vwoc:indicateOfLHType" minOccurs="0"/>
            <element name="HairLeftPart" type="vwoc:indicateOfLHType" minOccurs="0"/>
            <element name="HairPartsBangs" type="vwoc:indicateOfLHType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
    </complexType>

    <complexType name="EyebrowsType">
        <sequence>
            <element name="EyebrowSize" type="vwoc:indicateOfSMLType" minOccurs="0"/>
            <element name="EyebrowDensity" minOccurs="0">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="low"/>
                        <enumeration value="moderate"/>
                        <enumeration value="high"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="EyebrowHeight" type="vwoc:measureUnitLMHType" minOccurs="0"/>
            <element name="EyebrowArc" minOccurs="0">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="flat"/>
                        <enumeration value="middle"/>
                        <enumeration value="arched"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="EyebrowPoints" type="vwoc:indicateOfDMUType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
    </complexType>

    <complexType name="FacialHairType">
        <sequence>
            <element name="FacialHairThickness" type="vwoc:measureUnitLMHType" minOccurs="0"/>
```

```
            <element name="FacialSideburns" type="mpegvct:colorType" minOccurs="0"/>
            <element name="FacialMustache" type="boolean" minOccurs="0"/>
            <element name="FacialChinCurtains" type="boolean" minOccurs="0"/>
            <element name="FacialsoulPatch" type="boolean" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="hapticIDRef" type="IDREF" use="optional"/>
</complexType>

<complexType name="BodyHairType">
        <sequence>
            <element name="HairColor" type="mpegvct:colorType" minOccurs="0"/>
            <element name="HairThickness" type="vwoc:measureUnitLMHType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
</complexType>

<complexType name="FacialCalibrationPointsType">
        <sequence>
            <element name="Sellion" type="vwoc:PointType" minOccurs="0"/>
            <element name="RInfraorbitale" type="vwoc:PointType" minOccurs="0"/>
            <element name="LInfraorbitale" type="vwoc:PointType" minOccurs="0"/>
            <element name="Supramenton" type="vwoc:PointType" minOccurs="0"/>
            <element name="RTragion" type="vwoc:PointType" minOccurs="0"/>
            <element name="RGonion" type="vwoc:PointType" minOccurs="0"/>
            <element name="LTragion" type="vwoc:PointType" minOccurs="0"/>
            <element name="LGonion" type="vwoc:PointType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
</complexType>

<complexType name="PhysicalConditionType">
        <sequence>
            <element name="BodyStrength" type="vwoc:unlimitedPercentageType" minOccurs="0"/>
            <element name="BodyFlexibility" type="vwoc:indicateOfLMHType" minOccurs="0"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
</complexType>
```

5.2.4.2 Binary representation syntax

AvatarAppearanceType{	***Number of bits***	***Mnemonic***
BodyFlag	1	bslbf
HeadFlag	1	bslbf
EyesFlag	1	bslbf
EarsFlag	1	bslbf
NoseFlag	1	bslbf
MouthLipFlag	1	bslbf
BodySkinFlag	1	bslbf
FacialSkinFlag	1	bslbf
FacialFlag	1	bslbf

FingerNailsFlag	1	bslbf
ToeNailsFlag	1	bslbf
BodyLookFlag	1	bslbf
HairFlag	1	bslbf
EyeBrowsFlag	1	bslbf
FacialHairFlag	1	bslbf
BodyHairFlag	1	bslbf
AppearanceResourcesFlag	1	bslbf
FacialCalibrationPointsFlag	1	bslbf
PhysicalConditionFlag	1	bslbf
ClothesFlag	1	bslbf
ShoesFlag	1	bslbf
AccessoriesFlag	1	bslbf
SkinMarksFlag	1	bslbf
ExtraFlag	1	bslbf
if(BodyFlag){		
Body		BodyType
}		
if(HeadFlag){		
Head		HeadType
}		
if(EyesFlag){		
Eyes		EyesType
}		
if(EarsFlag){		
Ears		EarsType
}		
if(NoseFlag){		

Nose		NoseType
}		
if(MouthLipFlag){		
MouthLip		MouthLipType
}		
if(BodySkinFlag){		
BodySkin		SkinType
}		
if(FacialSkinFlag){		
FacialSkin		SkinType
}		
if(FacialFlag){		
Facial		FacialType
}		
if(FingerNailsFlag){		
FingerNails		NailType
}		
if(ToeNailsFlag){		
ToeNails		NailType
}		
if(BodyLookFlag){		
BodyLook		BodyLookType
}		
if(HairFlag){		
Hair		HairType
}		
if(EyeBrowsFlag){		
EyeBrows		EyeBrowsType
}		

if(FacialHairFlag){		
FacialHair		FacialHairType
}		
if(BodyHairFlag){		
BodyHair		BodyHairType
}		
if(AppearanceResourcesFlag){		
NumAppearanceResources		vluimsbf5
for(k=0; k< NumAppearanceResources; k++){		
AppearanceResources[k]	See ISO 10646	UTF-8
}		
}		
if(FacialCalibrationPointsFlag){		
FacialCalibrationPoints		FacialCalibrationPointsType
}		
if(PhysicalConditionFlag){		
PhysicalCondition		PhysicalConditionType
}		
if(ClothesFlag){		
NumClothes		vluimsbf5
for(k=0; k< NumClothes; k++){		
Clothes[k]		VirtualObjectType
}		
}		
if(ShoesFlag){		
NumShoes		vluimsbf5
for(k=0; k< NumShoes; k++){		

Shoes[k]		VirtualObjectType
}		
}		
if(AccessoriesFlag){		
NumAccessories		vluimsbf5
for(k=0; k< NumAccessories; k++){		
Accessories[k]		VirtualObjectType
}		
}		
if(SkinMarksFlag){		
NumSkinMarks		vluimsbf5
for(k=0; k< NumSkinMarks; k++){		
SkinMarks[k]		VirtualObjectType
}		
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
}		
BodyType{	***Number of bits***	***Mnemonic***
BodyHeightFlag	1	bslbf
BodyThicknessFlag	1	bslbf
BodyFatFlag	1	bslbf
TorsoMusclesFlag	1	bslbf

NeckThiknessFlag	1	bslbf
NeckLengthFlag	1	bslbf
ShouldersFlag	1	bslbf
PectorialsFlag	1	bslbf
ArmLengthFlag	1	bslbf
HeadSizeFlag	1	bslbf
TorsoLengthFlag	1	bslbf
LoveHandlesFlag	1	bslbf
BellySizeFlag	1	bslbf
LegMusclesFlag	1	bslbf
LegLengthFlag	1	bslbf
HipWidthFlag	1	bslbf
HipLengthFlag	1	bslbf
ButtSizeFlag	1	bslbf
PackageFlag	1	bslbf
SaddleBagsFlag	1	bslbf
KneeAngleFlag	1	bslbf
FootSizeFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(BodyHeightFlag) {		
BodyHeight	32	fsbf
}		
if(BodyThicknessFlag) {		
BodyThickness	32	fsbf
}		
if(BodyHeightFlag) {		
BodyFat		measureUnitLMHType

}		
if(TorsoMusclesFlag) {		
TorsoMuscles		measureUnitLMHType
}		
if(NeckThicknessFlag) {		
NeckThickness	32	fsbf
}		
if(NeckLengthFlag) {		
NeckLengthness	32	fsbf
}		
if(ShouldersFlag) {		
Shoulders	32	fsbf
}		
if(PectorialsFlag) {		
Pectorials	32	fsbf
}		
if(ArmLengthFlag) {		
ArmLength	32	fsbf
}		
if(HeadSizeFlag) {		
HeadSize	32	fsbf
}		
if(TorsoLengthFlag) {		
TorsoLength	32	fsbf
}		
if(LoveHandlesFlag) {		
LoveHandles	32	fsbf
}		
if(BellySizeFlag) {		

BellySize	32	fsbf
}		
if(LegMusclesFlag) {		
LegMuscles	32	fsbf
}		
if(LegLengthFlag) {		
LegLength	32	fsbf
}		
if(HipWidthFlag) {		
HipWidth	32	fsbf
}		
if(HipLengthFlag) {		
HipLength	32	fsbf
}		
if(ButtSizeFlag) {		
ButtSize	32	fsbf
}		
if(PackageFlag) {		
Package		indicateOfSMBType
}		
if(SaddleBagsFlag) {		
SaddleBags		indicateOfSMBType
}		
if(KneeAngleFlag) {		
KneeAngle		angleType
}		
if(FootSizeFlag) {		
FootSize	32	fsbf

}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag) {		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
HeadType{	***Number of bits***	***Mnemonic***
HeadSizeFlag	1	bslbf
HeadStrechFlag	1	bslbf
HeadShapeFlag	1	bslbf
EggHeadFlag	1	bslbf
HeadLengthFlag	1	bslbf
FaceShearFlag	1	bslbf
ForeheadSizeFlag	1	bslbf
ForeheadAngleFlag	1	bslbf
BrowSizeFlag	1	bslbf
FaceSkinFlag	1	bslbf
CheeksFlag	1	bslbf
CheeksDepthFlag	1	bslbf
CheeksShapeFlag	1	bslbf
UpperCheeksFlag	1	bslbf
LowerCheeksFlag	1	bslbf

CheekBonesFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(HeadSizeFlag){		
HeadSize		measureUnitSMBType
}		
if(HeadStrechFlag){		
HeadStrech	32	unlimitedPercentageType
}		
if(HeadShapeFlag){		
HeadShape	2	bslbf
}		
if(EggHeadFlag){		
EggHead	1	bslbf
}		
if(HeadLengthFlag){		
HeadLength	32	fsbf
}		
if(FaceShearFlag){		
FaceShear	32	fsbf
}		
if(ForeheadSizeFlag){		
ForeheadSize	32	fsbf
}		
if(ForeheadAngleFlag){		
ForeheadAngle		angleType
}		
if(BrowSizeFlag){		

BrowSize	32	fsbf
}		
if(FaceSkinFlag){		
FaceSkin	2	bslbf
}		
if(CheeksFlag){		
Cheeks		measureUnitSMBType
}		
if(CheeksDepthFlag){		
CheeksDepth	32	fsbf
}		
if(CheeksShapeFlag){		
CheeksShape	2	bslbf
}		
if(UpperCheeksFlag){		
UpperCheeks		measureUnitSMBType
}		
if(LowerCheeksFlag){		
LowerCheeks		measureUnitSMBType
}		
if(CheekBonesFlag){		
CheekBones		indicateOfDMUType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		

if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
EyesType{	***Number of bits***	***Mnemonic***
EyeSizeFlag	1	bslbf
EyeOpeningFlag	1	bslbf
EyeSpacingFlag	1	bslbf
OuterEyeCornerFlag	1	bslbf
InnerEyeCornerFlag	1	bslbf
EyeDepthFlag	1	bslbf
UpperEyelidFoldFlag	1	bslbf
EyeBagsFlag	1	bslbf
PuffyEyeLidsFlag	1	bslbf
EyelashLengthFlag	1	bslbf
EyePopFlag	1	bslbf
EyeColorFlag	1	bslbf
EyeLightnessFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(EyeSizeFlag){		
EyeSize	32	fsbf
}		
if(EyeOpeningFlag){		
EyeOpening		unlimitedPercentageType
}		

if(EyeSpacingFlag){		
EyeSpacing	32	fsbf
}		
if(OuterEyeCornerFlag){		
OuterEyeCorner		indicateOfDMUType
}		
if(InnerEyeCornerFlag){		
InnerEyeCorner		indicateOfDMUType
}		
if(EyeDepthFlag){		
EyeDepth	32	fsbf
}		
if(UpperEyelidFoldFlag){		
UpperEyelidFold	32	fsbf
}		
if(EyeBagsFlag){		
EyeBags	32	fsbf
}		
if(PuffyEyeLidsFlag){		
PuffyEyeLids		indicateOfSMBType
}		
if(EyelashLengthFlag){		
EyelashLength	32	fsbf
}		
if(EyePopFlag){		
EyePop	32	fsbf
}		
if(EyeColorFlag){		
EyeColor		colorType

}		
if(EyeLightnessFlag){		
EyeLightness		percentageType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
EarsType{	***Number of bits***	***Mnemonic***
EarSizeFlag	1	bslbf
EarPositionFlag	1	bslbf
EarAngleFlag	1	bslbf
AttachedEarlobesFlag	1	bslbf
EarTipsFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(EyeSizeFlag){		
EarSize	32	fsbf
}		
if(EarPositionFlag){		

EarPosition		indicateOfDMUType
}		
if(EarAngleFlag){		
EarAngle		angleType
}		
if(AttachedEarlobesFlag){		
AttachedEarlobes	32	fsbf
}		
if(EarTipsFlag){		
EarTips		indicateOfPMNType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
NoseType{	***Number of bits***	***Mnemonic***
NoseSizeFlag	1	bslbf
NoseWidthFlag	1	bslbf
NostrillWidthFlag	1	bslbf
NostrillDivisionFlag	1	bslbf
NoseThicknessFlag	1	bslbf

UpperBridgeFlag	1	bslbf
LowerBridgeFlag	1	bslbf
BridgeWidthFlag	1	bslbf
NoseTipAngleFlag	1	bslbf
NoseTipShapeFlag	1	bslbf
CrookedNoseFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(NoseSizeFlag){		
NoseSize	32	fsbf
}		
if(NoseWidthFlag){		
NoseWidth	32	fsbf
}		
if(NostrillWidthFlag){		
NostrillWidth	32	fsbf
}		
if(NostrillDivisionFlag){		
NostrillDivision	32	fsbf
}		
if(NoseThicknessFlag){		
NoseThickness	32	fsbf
}		
if(UpperBridgeFlag){		
UpperBridge	32	fsbf
}		
if(LowerBridgeFlag){		
LowerBridge	32	fsbf

}		
if(BridgeWidthFlag){		
BridgeWidth	32	fsbf
}		
if(NoseTipAngleFlag){		
NoseTipAngle		indicateOfDUType
}		
if(NoseTipShapeFlag){		
NoseTipShape	1	bslbf
}		
if(CrookedNoseFlag){		
CrookedNose		indicateOfLRType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
MouthLipType{	***Number of bits***	***Mnemonic***
LipWidthFlag	1	bslbf
LipFullnessFlag	1	bslbf
LipThicknessFlag	1	bslbf

LipRatioFlag	1	bslbf
MouthSizeFlag	1	bslbf
MouthPositionFlag	1	bslbf
MouthCornerFlag	1	bslbf
LipCleftDepthFlag	1	bslbf
LipCleftFlag	1	bslbf
ShiftMouthFlag	1	bslbf
ChinAngleFlag	1	bslbf
JawShapeFlag	1	bslbf
ChinDepthFlag	1	bslbf
JawAngleFlag	1	bslbf
JawJutFlag	1	bslbf
JowlsFlag	1	bslbf
ChinCleftFlag	1	bslbf
UpperChinCleftFlag	1	bslbf
ChinNeckFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(LipWidthFlag){		
LipWidth	32	fsbf
}		
if(LipFullnessFlag){		
LipFullness	32	fsbf
}		
if(LipThicknessFlag){		
LipThickness	32	fsbf
}		
if(LipRatioFlag){		

LipRatio	32	fsbf
}		
if(MouthSizeFlag){		
MouthSize	32	fsbf
}		
if(MouthPositionFlag){		
MouthPosition	32	fsbf
}		
if(MouthCornerFlag){		
MouthCorner		indicateOfDMUType
}		
if(LipCleftDepthFlag){		
LipCleftDepth	32	fsbf
}		
if(LipCleftFlag){		
LipCleft	32	fsbf
}		
if(ShiftMouthFlag){		
ShiftMouth		indicateOfLMRType
}		
if(ChinAngleFlag){		
ChinAngle	1	bslbf
}		
if(JawShapeFlag){		
JawShape		indicateOfPMNType
}		
if(ChinDepthFlag){		
ChinDepth	32	fsbf
}		

if(JawAngleFlag){		
JawAngle	32	fsbf
}		
if(JawJutFlag){		
JawJut	1	bslbf
}		
if(JowlsFlag){		
Jowls	32	fsbf
}		
if(ChinCleftFlag){		
ChinCleft		indicateOfRCType
}		
if(UpperChinCleftFlag){		
UpperChinCleft		indicateOfRCType
}		
if(ChinNeckFlag){		
ChinNeck	32	fsbf
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k<NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		

SkinType{	***Number of bits***	***Mnemonic***
SkinPigmentFlag	1	bslbf
SkinRuddinessFlag	1	bslbf
SkinRainbowColorFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(SkinPigmentFlag){		
SkinPigment		colorType
}		
if(SkinRuddinessFlag){		
SkinRuddiness		percentageType
}		
if(SkinRainbowColorFlag){		
SkinRainbowColor		colorType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		

FacialType{	*Number of bits*	*Mnemonic*
FacialDefinitionFlag	1	bslbf
FrecklesFlag	1	bslbf
WrinklesFlag	1	bslbf
RosyComplexionFlag	1	bslbf
LipPinknessFlag	1	bslbf
LipstickFlag	1	bslbf
LipstickColorFlag	1	bslbf
LipGlossFlag	1	bslbf
BlushFlag	1	bslbf
BlushColorFlag	1	bslbf
BlushOpacityFlag	1	bslbf
InnerShadowFlag	1	bslbf
InnerShadowColorFlag	1	bslbf
InnerShadowOppacityFlag	1	bslbf
OuterShadowFlag	1	bslbf
OuterShadowOppacityFlag	1	bslbf
EyeLinerFlag	1	bslbf
EyeLinerColorFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(FacialDefinitionFlag){		
FacialDefinition		levelOf5Type
}		
if(FrecklesFlag){		
Freckles		levelOf5Type
}		
if(WrinklesFlag){		

Wrinkles	1	bslbf
}		
if(RosyComplexionFlag){		
RosyComplexion	1	bslbf
}		
if(LipPinknessFlag){		
LipPinkness		levelOf5Type
}		
if(LipstickFlag){		
Lipstick	1	bslbf
}		
if(LipstickColorFlag){		
LipstickColor		colorType
}		
if(LipGlossFlag){		
LipGloss		levelOf5Type
}		
if(BlushFlag){		
Blush	1	bslbf
}		
if(BlushColorFlag){		
BlushColor		colorType
}		
if(BlushOpacityFlag){		
BlushOpacity		percentageType
}		
if(InnerShadowFlag){		
InnerShadow	1	bslbf
}		

if(InnerShadowColorFlag){		
InnerShadowColor		colorType
}		
if(InnerShadowOppacityFlag){		
InnerShadowOppacity		percentageType
}		
if(OuterShadowFlag){		
OuterShadow	1	bslbf
}		
if(OuterShadowOppacityFlag){		
OuterShadowOppacity		percentageType
}		
if(EyeLinerFlag){		
EyeLiner	1	bslbf
}		
if(EyeLinerColorFlag){		
EyeLinerColor		colorType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k<NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		

NailType{	***Number of bits***	***Mnemonic***
NailPolishFlag	1	bslbf
NailPolishColorFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(NailPolishFlag){		
NailPolish	1	bslbf
}		
if(NailPolishColorFlag){		
NailPolishColor		colorType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
BodyLookType{	***Number of bits***	***Mnemonic***
BodyDefinitionFlag	1	bslbf
BodyFrecklesFlag	1	bslbf
ExtraFlag	1	bslbf

if(BodyDefinitionFlag){		
BodyDefinition		indicateOfSMLType
}		
if(BodyFrecklesFlag){		
BodyFreckles		levelOf5Type
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
}		
HairType{	***Number of bits***	***Mnemonic***
HairSizeFlag	1	bslbf
HairStyleFlag	1	bslbf
HairColorFlag	1	bslbf
WhiteHairFlag	1	bslbf
RainbowColorFlag	1	bslbf
BlondeHairFlag	1	bslbf
RedHairFlag	1	bslbf
HairVolumeFlag	1	bslbf
HairFrontFlag	1	bslbf
HairSidesFlag	1	bslbf
HairBackFlag	1	bslbf
BigHairFrontFlag	1	bslbf
BigHairTopFlag	1	bslbf

BigHairBackFlag	1	bslbf
FrontFringeFlag	1	bslbf
SideFringeFlag	1	bslbf
BackFringeFlag	1	bslbf
FullHairSidesFlag	1	bslbf
HairSweepFlag	1	bslbf
ShearFrontFlag	1	bslbf
ShearBackFlag	1	bslbf
TuperFrontFlag	1	bslbf
TuperBackFlag	1	bslbf
RumpledhairFlag	1	bslbf
PigtailsFlag	1	bslbf
PonytailFlag	1	bslbf
SpikedHairFlag	1	bslbf
HairTiltFlag	1	bslbf
HairMiddlePartFlag	1	bslbf
HairRightPartFlag	1	bslbf
HairLeftPartFlag	1	bslbf
HairPartBangsFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(HairSizeFlag){		
HairSize		indicateOfSMLType
}		
if(HairStyleFlag){		
HairStyle		bslbf
}		
if(HairColorFlag){		
HairColor		colorType

}		
if(WhiteHairFlag){		
WhiteHair		percentageType
}		
if(RainbowColorFlag){		
RainbowColor		colorType
}		
if(BlondeHairFlag){		
BlondeHair		percentageType
}		
if(RedHairFlag){		
RedHair		percentageType
}		
if(HairVolumeFlag){		
HairVolume		indicateOfSMBType
}		
if(HairFrontFlag){		
HairFront		indicateOfSMLType
}		
if(HairSidesFlag){		
HairSides		indicateOfSMLType
}		
if(HairBackFlag){		
HairBack		indicateOfSMLType
}		
if(BigHairFrontFlag){		
BigHairFront		indicateOfSMLType
}		

if(BigHairTopFlag){		
BigHairTop		indicateOfSMLType
}		
if(BigHairBackFlag){		
BigHairBack		indicateOfSMLType
}		
if(FrontFringeFlag){		
FrontFringe		indicateOfSMLType
}		
if(SideFringeFlag){		
SideFringe		indicateOfSMLType
}		
if(BackFringeFlag){		
BackFringe		indicateOfSMLType
}		
if(FullHairSidesFlag){		
FullHairSides		indicateOfSMLType
}		
if(HairSweepFlag){		
HairSweep		indicateOfSMLType
}		
if(ShearFrontFlag){		
ShearFront		indicateOfLMRType
}		
if(ShearBackFlag){		
ShearBack		indicateOfSMLType
}		
if(TuperFrontFlag){		
TuperFront		indicateOfSMLType

}		
if(TuperBackFlag){		
TuperBack		indicateOfSMLType
}		
if(RumpledhairFlag){		
Rumpledhair	2	bslbf
}		
if(PigtailsFlag){		
Pigtails		indicateOfSMLType
}		
if(PonytailFlag){		
Ponytail		indicateOfSMLType
}		
if(SpikedHairFlag){		
SpikedHair		indicateOfSMLType
}		
if(HairTiltFlag){		
HairTilt	32	fsbf
}		
if(HairMiddlePartFlag){		
HairMiddlePart		indicateOfLHType
}		
if(HairRightPartFlag){		
HairRightPart		indicateOfLHType
}		
if(HairLeftPartFlag){		
HairLeftPart		indicateOfLHType
}		

if(HairPartBangsFlag){		
HairPartBangs		indicateOfLHType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
EyebrowsType{	***Number of bits***	***Mnemonic***
EyebrowSizeFlag	1	bslbf
EyebrowDensityFlag	1	bslbf
EyebrowHeightFlag	1	bslbf
EyebrowArcFlag	1	bslbf
EyebrowPointsFlag	1	bslbf
ExtraFlag	1	bslbf
hapticIDRefFlag	1	bslbf
if(EyebrowSizeFlag){		
EyebrowSize		indicateOfSMLType
}		
if(EyebrowDensityFlag){		
EyebrowDensity	2	bslbf
}		

if(EyebrowHeightFlag){		
EyebrowHeight		measureUnitLMHType
}		
if(EyebrowArcFlag){		
EyebrowArc	2	bslbf
}		
if(EyebrowPointsFlag){		
EyebrowPoints		indicateOfDMUType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
FacialHairType{	***Number of bits***	***Mnemonic***
FacialHairThicknessFlag	1	bslbf
FacialSideburnsFlag	1	bslbf
FacialMustacheFlag	1	bslbf
FacialChinCurtainsFlag	1	bslbf
FacialSoulPatchFlag	1	bslbf
ExtraFlag	1	bslbf

hapticIDRefFlag	1	bslbf
if(FacialHairThicknessFlag){		
FacialHairThickness		measureUnitLMHType
}		
if(FacialSideburnsFlag){		
FacialSideburns		colorType
}		
if(FacialMustacheFlag){		
FacialMustache	1	bslbf
}		
if(FacialChinCurtainsFlag){		
FacialChinCurtains	1	bslbf
}		
if(FacialSoulPatchFlag){		
FacialSoulPatch	1	bslbf
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
if(hapticIDRefFlag){		
hapticIDRef	See ISO 10646	UTF-8
}		
}		
BodyHairType {	***Number of bits***	***Mnemonic***

HairColorFlag	1	bslbf
HairThicknessFlag	1	bslbf
ExtraFlag	1	bslbf
if(HairColorFlag) {		
HairColor		colorType
}		
if(HairThicknessFlag) {		
HairThickness		measureUnitLMHType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
}		
FacialCalibrationPointsType{	***Number of bits***	***Mnemonic***
SellionFlag	1	bslbf
RInfraorbitaleFlag	1	bslbf
LinfraorbitaleFlag	1	bslbf
SupramentonFlag	1	bslbf
RtragionFlag	1	bslbf
RgonionFlag	1	bslbf
LtragionFlag	1	bslbf
LgonionFlag	1	bslbf
ExtraFlag	1	bslbf
if(SellionFlag){		

Sellion		PointType
}		
if(RinfraorbitaleFlag){		
Rinfraorbitale		PointType
}		
if(LinfraorbitaleFlag){		
Linfraorbitale		PointType
}		
if(SupramentonFlag){		
Supramenton		PointType
}		
if(RtragionFlag){		
Rtragion		PointType
}		
if(RgonionFlag){		
Rgonion		PointType
}		
if(LtragionFlag){		
Ltragion		PointType
}		
if(LgonionFlag){		
Lgonion		PointType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		

}		
PhysicalConditionType{	***Number of bits***	***Mnemonic***
BodyStrengthFlag	1	bslbf
BodyFlexibilityFlag	1	bslbf
ExtraFlag	1	bslbf
if(BodyStrengthFlag){		
BodyStrength		unlimitedPercentageType
}		
if(BodyFlexibilityFlag){		
BodyFlexibility		indicateOfLMHType
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
}		

5.2.4.3 Semantics

Name	*Description*
`AvatarAppearanceType`	A type that contains the high-level description of the avatar appearance and may refer a media containing the exact geometry and texture.
`BodyFlag`	This field, which is only present in the binary representation, signals the presence of the `Body` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`HeadFlag`	This field, which is only present in the binary representation, signals the presence of the `Head` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`EyesFlag`	This field, which is only present in the binary representation, signals the presence of the `Eyes` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`EarsFlag`	This field, which is only present in the binary representation, signals the presence of the `Ears`

	element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
NoseFlag	This field, which is only present in the binary representation, signals the presence of the Nose element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
MouthLipFlag	This field, which is only present in the binary representation, signals the presence of the Mouthlip element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BodySkinFlag	This field, which is only present in the binary representation, signals the presence of the BodySkin element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FacialSkinFlag	This field, which is only present in the binary representation, signals the presence of the FacialSkin element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FacialFlag	This field, which is only present in the binary representation, signals the presence of the facial element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FingerNailsFlag	This field, which is only present in the binary representation, signals the presence of the FingerNails element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ToeNailsFlag	This field, which is only present in the binary representation, signals the presence of the ToeNails element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BodyLookFlag	This field, which is only present in the binary representation, signals the presence of the BodyLook element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairFlag	This field, which is only present in the binary representation, signals the presence of the Hair element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeBrowsFlag	This field, which is only present in the binary representation, signals the presence of the Eyebrows element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FacialHairFlag	This field, which is only present in the binary representation, signals the presence of the FacialHair element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BodyHairFlag	This field, which is only present in the binary representation, signals the presence of the BodyHair element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
AppearanceResourcesFlag	This field, which is only present in the binary representation, signals the presence of the AppearanceResource element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FacialCalibrationPointsFlag	This field, which is only present in the binary representation, signals the presence of the FacialCalibrationPoints element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
PhysicalConditionFlag	This field, which is only present in the binary representation, signals the presence of the PhysicalCondition element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ClothesFlag	This field, which is only present in the binary representation, signals the presence of the clothes element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ShoesFlag	This field, which is only present in the binary representation, signals the presence of the shoes element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
AccessoriesFlag	This field, which is only present in the binary representation, signals the presence of the accessories element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
SkinMarksFlag	This field, which is only present in the binary representation, signals the presence of the SkinMarks element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation, signals the presence of the

<table>
<tr><td></td><td><code>extra</code> type element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>Body</code></td><td>Set of descriptions for body of the avatar.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td><code>BodyType</code></td><td>A type that describes avatar body.</td></tr>
<tr><td><code>BodyHeightFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>BodyHeight</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>BodyThicknessFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>BodyThickness</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>BodyFatFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>BodyFat</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>TorsoMusclesFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>TorsoMuscles</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>NeckThiknessFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>NeckThikness</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>NeckLengthFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>NeckLength</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>ShouldersFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>Shoulders</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>PectorialsFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>Pectorials</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>ArmLengthFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>ArmLength</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>HeadSizeFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>HeadSize</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>TorsoLengthFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>TorsoLength</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>LoveHandlesFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>LoveHandles</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>BellySizeFlag</code></td><td>This field, which is only present in the binary representation, signals the presence of the <code>BellySize</code> element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td><code>LegMusclesFlag</code></td><td>This field, which is only present in the binary representation,</td></tr>
</table></td></tr>
</table>

	signals the presence of the LegMuscles element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
LegLengthFlag	This field, which is only present in the binary representation, signals the presence of the LegLength element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HipWidthFlag	This field, which is only present in the binary representation, signals the presence of the HipWidth element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HipLengthFlag	This field, which is only present in the binary representation, signals the presence of the HipLength element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ButtSizeFlag	This field, which is only present in the binary representation, signals the presence of the ButtSize element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
PackageFlag	This field, which is only present in the binary representation, signals the presence of the Package element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
SaddleBagsFlag	This field, which is only present in the binary representation, signals the presence of the SaddleBags element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
KneeAngleFlag	This field, which is only present in the binary representation, signals the presence of the KneeAngle element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FootSizeFlag	This field, which is only present in the binary representation, signals the presence of the FootSize element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of hapticIDRef of the body type element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BodyHeight	Full height of the character (always in meter).
BodyThickness	This indicates the weight of the bounding box of the avatar (always in meter).
BodyFat	This should be one of Low, Medium, High and indicates the fatness of the body.
TorsoMuscles	This should be one of Low, Medium, High and indicates the average muscularity of the avatar's body.
NeckThikness	The diameter of the neck (always in meter).
NeckLength	The height of the neck (always in meter).
Shoulders	The width of the shoulders (always in meter).
Pectorials	The size of the pectoral muscles (always in meter).
ArmLength	Length of complete arm (always in meter).
HandSize	Size of the whole hand including fingers (always in meter).
TorsoLength	The length of torso (between pectorals and legs) (always in meter)
LoveHandles	Size of the love handles (always in meter).

	BellySize	Diameter of the belly (always in meter).
	LegMucles	Size of all leg muscles (always in meter).
	LegLength	Length of complete leg (always in meter).
	HipWidth	The width of the hip area (always in meter).
	HipLength	The vertical size of the hip area (always in meter).
	ButtSize	Diameter of the butt's avatar (always in meter).
	Package	Size of the package (small, medium, big).
	SaddleBags	Volume of saddle bags (small, medium, big).
	KneeAngle	The angle between the upper end lower leg, normally 0 when they are aligned (in degrees, from 0 to 360).
	FootSize	Size of the whole foot including toes (always in meter).
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the BodyType.
	Extra	Describes any other descriptions of body.
	hapticIDRef	Identifier that refers to the haptic properties of the body.
Head	Set of descriptions for head of the avatar.	
	Name	*Description*
	HeadType	A type that describes avatar head.
	HeadSizeFlag	This field, which is only present in the binary representation, signals the presence of the HeadSize element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	HeadStrechFlag	This field, which is only present in the binary representation, signals the presence of the HeadStrech element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	HeadShapeFlag	This field, which is only present in the binary representation, signals the presence of the HeadShape element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	EggHeadFlag	This field, which is only present in the binary representation, signals the presence of the EggHead element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	HeadLengthFlag	This field, which is only present in the binary representation, signals the presence of the HeadLength element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	FaceShearFlag	This field, which is only present in the binary representation, signals the presence of the FaceShear element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ForeheadSizeFlag	This field, which is only present in the binary representation, signals the presence of the ForeheadSize element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ForeheadAngleFlag	This field, which is only present in the binary representation, signals the presence of the ForeheadAngle element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	BrowSizeFlag	This field, which is only present in the binary representation, signals the presence of the BrowSize element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	FaceSkinFlag	This field, which is only present in the binary representation, signals the presence of the FaceSkin element. ”1” means that the element shall be used. ”0” means that the element shall not

	be used.
CheeksFlag	This field, which is only present in the binary representation, signals the presence of the Cheeks element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
CheeksDepthFlag	This field, which is only present in the binary representation, signals the presence of the CheeksDepth element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
CheeksShapeFlag	This field, which is only present in the binary representation, signals the presence of the CheeksShape element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
UpperCheeksFlag	This field, which is only present in the binary representation, signals the presence of the UpperCheeks element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
LowerCheeksFlag	This field, which is only present in the binary representation, signals the presence of the LowerCheeks element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
CheekBonesFlag	This field, which is only present in the binary representation, signals the presence of the CheekBones element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
HeadSize	Size of the entire head (small, medium, big).
HeadStrech	Vertical stretch of the head in %.
HeadShape	This can be one of "square", "round", “oval”, or "long”.
EggHead	Head is larger on the top than on the bottom or vice versa. This can be "yes" or "not".
HeadLength	The distance between the face and the back of the head, flat head or long head, measured in metres.
FaceShear	Changes the height difference between the two sides of the face (always in metre).
ForeheadSize	The height of the forehead measured in metres.
ForeheadAngle	The angle of the forehead measured in degrees.
BrowSize	Measures how much the eyebrows are extruded from the face (in metre).
FaceSkin	Describe the type of face skin (dry, normal, greasy).
Cheeks	The size of the complete cheeks (small, medium, big).
CheeksDepth	The depth of the complete cheeks (always in metre).
CheeksShape	Different cheeks shapes (one of the following values: chubby, high, bone)
UpperCheeks	The volume of the upper cheeks (small, medium, big).
LowerCheeks	The volume of the lower cheeks (small, medium, big).
CheekBones	The vertical position of the cheek bones (down, medium, up).
NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the HeadType.
Extra	Describes any other descriptions of head.
hapticIDRef	Identifier that refers to the haptic properties of the head.

Eyes	Set of descriptions for eyes of the avatar.

Name	*Description*
EyesType	A type that describes avatar eyes.
EyeSizeFlag	This field, which is only present in the binary representation, signals the presence of the EyeSize element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeOpeningFlag	This field, which is only present in the binary representation, signals the presence of the EyeOpening element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeSpacingFlag	This field, which is only present in the binary representation, signals the presence of the EyeSpacing element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
OuterEyeCornerFlag	This field, which is only present in the binary representation, signals the presence of the OuterEyeCorner element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
InnerEyeCornerFlag	This field, which is only present in the binary representation, signals the presence of the InnerEyeCorner element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeDepthFlag	This field, which is only present in the binary representation, signals the presence of the EyeDepth element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
UpperEyelidFoldFlag	This field, which is only present in the binary representation, signals the presence of the UpperEyelidFold element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeBagsFlag	This field, which is only present in the binary representation, signals the presence of the EyeBags element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
PuffyEyeLidsFlag	This field, which is only present in the binary representation, signals the presence of the PuffyEyeLids element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyelashLengthFlag	This field, which is only present in the binary representation, signals the presence of the EyelashLength element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyePopFlag	This field, which is only present in the binary representation, signals the presence of the EyePop element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeColorFlag	This field, which is only present in the binary representation, signals the presence of the EyeColor element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeLightnessFlag	This field, which is only present in the binary representation, signals the presence of the EyeLightness element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means

		that the element shall be used. "0" means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. "1" means that the attribute shall be used. "0" means that the attribute shall not be used.
	EyeSize	The size of the entire eyes (always in metre).
	EyeOpening	How much the eyelids are opened (always in metre).
	EyeSpacing	Distance between the eyes (always in metre).
	OuterEyeCorner	Vertical position of the outer eye corner (down, middle, up).
	InnerEyeCorner	Vertical position of the inner eye corner (down, middle, up).
	EyeDepth	How much the eyes are inside the head (always in metre).
	UpperEyelidFold	How much the upper eyelid covers the eye (always in metre).
	EyeBags	The size of the eye bags (always in metre).
	PuffyEyelids	The volume of the eye bags (small, medium, big).
	EyelashLength	The length of the eyelashes (always in meter).
	EyePop	The size difference between the left and right eye (always in metre).
	EyeColor	The color type defined in ISO/IEC 23005-6 shall be used for eye colour.
	EyeLightness	The reflectivity of the eye in %.
	Extra	Describes any other descriptions of eyes.
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the EyesType.
	hapticIDRef	Identifier that refers to the haptic properties of the eyes.
Ears	Set of descriptions for ears of the avatar.	
	Name	*Description*
	EarsType	A type that describes avatar ears.
	EarSizeFlag	This field, which is only present in the binary representation, signals the presence of the EarSize element. "1" means that the element shall be used. "0" means that the element shall not be used.
	EarPositionFlag	This field, which is only present in the binary representation, signals the presence of the EarPosition element. "1" means that the element shall be used. "0" means that the element shall not be used.
	EarAngleFlag	This field, which is only present in the binary representation, signals the presence of the EarAngle element. "1" means that the element shall be used. "0" means that the element shall not be used.
	AttachedEarlobesFlag	This field, which is only present in the binary representation, signals the presence of the AttachedEarlobes element. "1" means that the element shall be used. "0" means that the element shall not be used.
	EarTipsFlag	This field, which is only present in the binary representation, signals the presence of the EarTips element. "1" means that the element shall be used. "0" means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. "1" means that the element shall be used. "0" means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. "1" means that the attribute shall be used. "0" means that the attribute shall not be used.
	EarSize	Size of the entire ear (always in metre).

	EarPosition	Vertical ear position on the head (down, middle, up).
	EarAngle	The angle between the ear and the head in degrees.
	AttachedEarlobes	The size of the earlobes (always in metre).
	EarTips	How much the ear tips are pointed (pointed, medium, not pointed).
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the EarsType.
	Extra	Describes any other descriptions of ears.
	hapticIDRef	Identifier that refers to the haptic properties of the ears.
Nose	Set of descriptions for nose of the avatar.	

Name	*Description*
NoseType	A type that describes avatar nose.
NoseSizeFlag	This field, which is only present in the binary representation, signals the presence of the NoseSize element. "1" means that the element shall be used. "0" means that the element shall not be used.
NoseWidthFlag	This field, which is only present in the binary representation, signals the presence of the NoseWidth element. "1" means that the element shall be used. "0" means that the element shall not be used.
NostrillWidthFlag	This field, which is only present in the binary representation, signals the presence of the NostrillWidth element. "1" means that the element shall be used. "0" means that the element shall not be used.
NostrillDivisionFlag	This field, which is only present in the binary representation, signals the presence of the NostrillDivision element. "1" means that the element shall be used. "0" means that the element shall not be used.
NoseThicknessFlag	This field, which is only present in the binary representation, signals the presence of the NoseThickness element. "1" means that the element shall be used. "0" means that the element shall not be used.
UpperBridgeFlag	This field, which is only present in the binary representation, signals the presence of the UpperBridge element. "1" means that the element shall be used. "0" means that the element shall not be used.
LowerBridgeFlag	This field, which is only present in the binary representation, signals the presence of the LowerBridge element. "1" means that the element shall be used. "0" means that the element shall not be used.
BridgeWidthFlag	This field, which is only present in the binary representation, signals the presence of the BridgeWidth element. "1" means that the element shall be used. "0" means that the element shall not be used.
NoseTipAngleFlag	This field, which is only present in the binary representation, signals the presence of the NoseTipAngle element. "1" means that the element shall be used. "0" means that the element shall not be used.
NoseTipShapeFlag	This field, which is only present in the binary representation, signals the presence of the NoseTipShape element. "1" means that the element shall be used. "0" means that the element shall not be used.
CrookedNoseFlag	This field, which is only present in the binary representation, signals the presence of the CrookedNose element. "1" means that the element shall be used. "0" means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation,

		signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
	NoseSize	The height of the nose from its bottom (always in metre).
	NoseWidth	The width of the complete nose (always in metre).
	NostrillWidth	Width of only the nostrils (always in metre).
	NostrillDivision	The size of the nostril division (always in metre).
	NoseThickness	The size of the tip of the nose (always in metre).
	UpperBridge	The height of the upper part of the nose (always in metre).
	LowerBridge	The height of the lower part of the nose (always in metre).
	BridgeWidth	The width of the upper part of the nose (always in metre).
	NoseTipAngle	The angle of the nose tip, "up" or "down".
	NoseTipShape	The shape of the nose tip, "pointy" or "bulbous".
	CrookedNose	Displacement of the nose on the left or right side.
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the NoseType.
	Extra	Describes any other descriptions of nose.
	hapticIDRef	Identifier that refers to the haptic properties of the nose.
MouthLip	Set of descriptions for mouth and lips of the avatar.	
	Name	*Description*
	MouthLipType	A type that describes avatar eyes.
	LipWidthFlag	This field, which is only present in the binary representation, signals the presence of the LipWidth element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	LipFullnessFlag	This field, which is only present in the binary representation, signals the presence of the LipFullness element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	LipThicknessFlag	This field, which is only present in the binary representation, signals the presence of the LipThickness element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	LipRatioFlag	This field, which is only present in the binary representation, signals the presence of the LipRatio element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	MouthSizeFlag	This field, which is only present in the binary representation, signals the presence of the MouthSize element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	MouthPositionFlag	This field, which is only present in the binary representation, signals the presence of the MouthPosition element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	MouthCornerFlag	This field, which is only present in the binary representation, signals the presence of the MouthCorner element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	LipCleftDepthFlag	This field, which is only present in the binary representation, signals the presence of the LipCleftDepth element. ”1” means that the element shall be used. ”0” means that the element shall not be used.

	LipCleftFlag	This field, which is only present in the binary representation, signals the presence of the LipCleft element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ShiftMouthFlag	This field, which is only present in the binary representation, signals the presence of the ShiftMouth element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ChinAngleFlag	This field, which is only present in the binary representation, signals the presence of the ChinAngle element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	JawShapeFlag	This field, which is only present in the binary representation, signals the presence of the JawShape element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ChinDepthFlag	This field, which is only present in the binary representation, signals the presence of the ChinDepth element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	JawAngleFlag	This field, which is only present in the binary representation, signals the presence of the JawAngle element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	JawJutFlag	This field, which is only present in the binary representation, signals the presence of the JawJut element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	JowlsFlag	This field, which is only present in the binary representation, signals the presence of the Jowls element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ChinCleftFlag	This field, which is only present in the binary representation, signals the presence of the ChinCleft element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	UpperChinCleftFlag	This field, which is only present in the binary representation, signals the presence of the UpperChinCleft element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ChinNeckFlag	This field, which is only present in the binary representation, signals the presence of the ChinNeck element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
	LipWidth	The width of the lips (m).
	LipFullness	The fullness of the lip (m).
	LipThickness	The thickness of the lip (m).
	LipRatio	Difference between the upper and lower lip (m).
	MouthSize	The size of the complete mouth (m).
	MouthPosition	Vertical position of the mouth on the face (m).

	MouthCorner	Vertical position of the mouth corner (down, middle, up).
	LipCleftDepth	The height of the lip cleft (m).
	LipCleft	The width of the lip cleft (m).
	ShiftMouth	Horizontal position of mouth on the face (left, middle, right).
	ChinAngle	The curvature of the chin, outer or inner.
	JawShape	Pointy to Square jaw (pointed, middle, not pointed).
	ChinDepth	Vertical height of the chin (m).
	JawAngle	The height of the jaw (m).
	JawJut	Position of the jaw inside or out of the face (inside, outside).
	Jowls	The size of the jowls (m).
	ChinCleft	The shape of the chin cleft, "round" or "cleft".
	UpperChinCleft	The shape of the upper chin cleft, "round" or "cleft".
	ChinNeck	The size of the chin neck (m).
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the MouthLipType.
	Extra	Describes any other descriptions of mouthlip.
	hapticIDRef	Identifier that refers to the haptic properties of the mouth and lips.
BodySkin,	Set of descriptions for body skin of the avatar.	
	Name	*Description*
	SkinType	A type that describes avatar skin.
	SkinPigmentFlag	This field, which is only present in the binary representation, signals the presence of the SkinPigment element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	SkinRuddinessFlag	This field, which is only present in the binary representation, signals the presence of the SkinRuddiness element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	SkinRainbowColorFlag	This field, which is only present in the binary representation, signals the presence of the SkinRainbowColor element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
	SkinPigment	Body skin pigment (very light, light, average, olive, brown, black).
	SkinRuddiness	Body skin ruddiness (few, medium, lot).
	SkinRainbowColor	The color type defined in ISO/IEC 23005-6 shall be used for body skin rainbow colour.
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the BodySkinType.
	Extra	Describes any other descriptions of body skin.
	hapticIDRef	Identifier that refers to the haptic properties of the body skin.
FacialSkin	Set of descriptions for facial skin of the avatar.	
	Name	*Description*
	SkinType	A type that describes avatar skin.
	SkinPigmentFlag	This field, which is only present in the binary representation,

		signals the presence of the SkinPigment element. "1" means that the element shall be used. "0" means that the element shall not be used.
	SkinRuddinessFlag	This field, which is only present in the binary representation, signals the presence of the SkinRuddiness element. "1" means that the element shall be used. "0" means that the element shall not be used.
	SkinRainbowColorFlag	This field, which is only present in the binary representation, signals the presence of the SkinRainbowColor element. "1" means that the element shall be used. "0" means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. "1" means that the element shall be used. "0" means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. "1" means that the attribute shall be used. "0" means that the attribute shall not be used.
	SkinPigment	Facial skin pigment (very light, light, average, olive, brown, black).
	SkinRuddiness	Facial skin ruddiness (few, medium, lot).
	SkinRainbowColor	The color type defined in ISO/IEC 23005-6 shall be used for facial skin rainbow colour.
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the FacialSkinType.
	Extra	Describes any other descriptions of facial skin.
	hapticIDRef	Identifier that refers to the haptic properties of the skin.
Facial	Set of descriptions for face of the avatar.	
	Name	*Description*
	FacialType	A type that describes avatar face.
	FacialDefinitionFlag	This field, which is only present in the binary representation, signals the presence of the FacialDefinition element. "1" means that the element shall be used. "0" means that the element shall not be used.
	FrecklesFlag	This field, which is only present in the binary representation, signals the presence of the Freckles element. "1" means that the element shall be used. "0" means that the element shall not be used.
	WrinklesFlag	This field, which is only present in the binary representation, signals the presence of the Wrinkles element. "1" means that the element shall be used. "0" means that the element shall not be used.
	RosyComplexionFlag	This field, which is only present in the binary representation, signals the presence of the RosyComplexion element. "1" means that the element shall be used. "0" means that the element shall not be used.
	LipPinknessFlag	This field, which is only present in the binary representation, signals the presence of the LipPinkness element. "1" means that the element shall be used. "0" means that the element shall not be used.
	LipstickFlag	This field, which is only present in the binary representation, signals the presence of the Lipstick element. "1" means that the element shall be used. "0" means that the element shall not be used.
	LipstickColorFlag	This field, which is only present in the binary representation, signals the presence of the LipstickColor element. "1" means that the element shall be used. "0" means that the

	element shall not be used.
LipGlossFlag	This field, which is only present in the binary representation, signals the presence of the LipGloss element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BlushFlag	This field, which is only present in the binary representation, signals the presence of the Blush element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BlushColorFlag	This field, which is only present in the binary representation, signals the presence of the BlushColor element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BlushOpacityFlag	This field, which is only present in the binary representation, signals the presence of the BlushOpacity element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
InnerShadowFlag	This field, which is only present in the binary representation, signals the presence of the InnerShadow element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
InnerShadowColorFlag	This field, which is only present in the binary representation, signals the presence of the InnerShadowColor element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
InnerShadowOppacityFlag	This field, which is only present in the binary representation, signals the presence of the InnerShadowOppacity element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
OuterShadowFlag	This field, which is only present in the binary representation, signals the presence of the OuterShadow element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
OuterShadowOppacityFlag	This field, which is only present in the binary representation, signals the presence of the OuterShadowOppacity element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeLinerFlag	This field, which is only present in the binary representation, signals the presence of the EyeLiner element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
EyeLinerColorFlag	This field, which is only present in the binary representation, signals the presence of the EyeLinerColor element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
FacialDefinition	Level of brightness of the face from 1-lighted to 5 dark.
Freckles	Freckles (5 levels, 1=smallest, 5= biggest).
Wrinkles	Wrinkles (yes or no).
RosyComplexion	Rosy Complexion (yes or no).
LipPinkness	Lip Pinkness (5 levels, 1=smallest, 5= biggest).
Lipstick	Lipstick (yes or no).

	LipstickColor	The color type defined in ISO/IEC 23005-6 shall be used for lipstick colour.
	Lipgloss	Lipgloss (5 levels, 1=smallest, 5= biggest).
	Blush	Blush (yes or no).
	BlushColor	The color type defined in ISO/IEC 23005-6 shall be used for blush colour.
	BlushOpacity	Blush Opacity (%).
	InnerShadow	Inner Shadow (yes or no).
	InnerShadowColor	The color type defined in ISO/IEC 23005-6 shall be used for inner shadow colour.
	InnerShadowOpacity	Inner Shadow Opacity (%).
	OuterShadow	Outer Shadow (yes or no).
	OuterShadowOpacity	Outer Shadow Opacity (%).
	Eyeliner	Eyeliner (yes or no).
	EyelinerColor	The color type defined in ISO/IEC 23005-6 shall be used for eyeliner colour.
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the FacialType.
	Extra	Describes any other descriptions of face.
	hapticIDRef	Identifier that refers to the haptic properties of the face.
FingerNails,	Set of descriptions for finger nails of the avatar.	
	Name	*Description*
	NailType	A type that describes avatar nail.
	NailPolishFlag	This field, which is only present in the binary representation, signals the presence of the NailPolish element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	NailPolishColorFlag	This field, which is only present in the binary representation, signals the presence of the NailPolishColor element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
	NailPolish	Finger nail polish (yes or no).
	NailPolishColor	The color type defined in ISO/IEC 23005-6 shall be used for finger nail polish colour.
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the NailsType.
	Extra	Describes any other descriptions of finger nails.
	hapticIDRef	Identifier that refers to the haptic properties of the nails.
ToeNails	Set of descriptions for toe nails of the avatar.	
	Name	*Description*
	NailType	A type that describes avatar nail.
	NailPolishFlag	This field, which is only present in the binary representation, signals the presence of the NailPolish element. ”1” means that the element shall be used. ”0” means that the element shall not be used.

	NailPolishColorFlag	This field, which is only present in the binary representation, signals the presence of the NailPolishColor element. "1" means that the element shall be used. "0" means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. "1" means that the element shall be used. "0" means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. "1" means that the attribute shall be used. "0" means that the attribute shall not be used.
	NailPolish	Toe nail polish (yes or no)
	NailPolishColor	The color type defined in ISO/IEC 23005-6 shall be used for toe nail polish colour.
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the NailsType.
	Extra	Describes any other descriptions of toe nails.
	hapticIDRef	Identifier that refers to the haptic properties of the nails.
BodyLook	Set of descriptions for body look of the avatar.	
	Name	*Description*
	BodyLookType	A type that describes avatar body look.
	BodyDefinitionFlag	This field, which is only present in the binary representation, signals the presence of the BodyDefinition element. "1" means that the element shall be used. "0" means that the element shall not be used.
	BodyFrecklesFlag	This field, which is only present in the binary representation, signals the presence of the BodyFreckles element. "1" means that the element shall be used. "0" means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. "1" means that the element shall be used. "0" means that the element shall not be used.
	BodyDefinition	Body definition (small, medium, large)
	BodyFreckles	Body Freckles (5 levels, 1=smallest, 5= biggest)
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the BodyLookType.
	Extra	Describes any other descriptions of bodylook.
Hair	Set of elements for general avatar hair description. Containing elements:	
	Name	*Description*
	HairType	A type that describes avatar hair.
	HairSizeFlag	This field, which is only present in the binary representation, signals the presence of the HairSize element. "1" means that the element shall be used. "0" means that the element shall not be used.
	HairStyleFlag	This field, which is only present in the binary representation, signals the presence of the HairStyle element. "1" means that the element shall be used. "0" means that the element shall not be used.
	HairColorFlag	This field, which is only present in the binary representation, signals the presence of the HairColor element. "1" means that the element shall be used. "0" means that the element shall not be used.
	WhiteHairFlag	This field, which is only present in the binary representation, signals the presence of the WhiteHair element. "1" means that the element shall be used. "0" means that the element shall not

	be used.
RainbowColorFlag	This field, which is only present in the binary representation, signals the presence of the RainbowColor element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BlondeHairFlag	This field, which is only present in the binary representation, signals the presence of the BlondeHair element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
RedHairFlag	This field, which is only present in the binary representation, signals the presence of the RedHair element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairVolumeFlag	This field, which is only present in the binary representation, signals the presence of the HairVolume element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairFrontFlag	This field, which is only present in the binary representation, signals the presence of the HairFront element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairSidesFlag	This field, which is only present in the binary representation, signals the presence of the HairSides element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairBackFlag	This field, which is only present in the binary representation, signals the presence of the HairBack element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BigHairFrontFlag	This field, which is only present in the binary representation, signals the presence of the BigHairFront element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BigHairTopFlag	This field, which is only present in the binary representation, signals the presence of the BigHairTop element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BigHairBackFlag	This field, which is only present in the binary representation, signals the presence of the BigHairBack element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FrontFringeFlag	This field, which is only present in the binary representation, signals the presence of the FrontFringe element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
SideFringeFlag	This field, which is only present in the binary representation, signals the presence of the SideFringeFlag element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BackFringeFlag	This field, which is only present in the binary representation, signals the presence of the BackFringe element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FullHairSidesFlag	This field, which is only present in the binary representation, signals the presence of the FullHairSides element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairSweepFlag	This field, which is only present in the binary representation, signals the presence of the HairSweep element. ”1” means that

	the element shall be used. ”0” means that the element shall not be used.
ShearFrontFlag	This field, which is only present in the binary representation, signals the presence of the ShearFront element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ShearBackFlag	This field, which is only present in the binary representation, signals the presence of the ShearBack element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
TuperFrontFlag	This field, which is only present in the binary representation, signals the presence of the TuperFront element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
TuperBackFlag	This field, which is only present in the binary representation, signals the presence of the TuperBack element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
RumpledhairFlag	This field, which is only present in the binary representation, signals the presence of the RumpledhairFlag element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
PigtailsFlag	This field, which is only present in the binary representation, signals the presence of the Pigtails element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
PonytailFlag	This field, which is only present in the binary representation, signals the presence of the Ponytail element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
SpikedHairFlag	This field, which is only present in the binary representation, signals the presence of the SpikedHair element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairTiltFlag	This field, which is only present in the binary representation, signals the presence of the HairTilt element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairMiddlePartFlag	This field, which is only present in the binary representation, signals the presence of the HairMiddlePart element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairRightPartFlag	This field, which is only present in the binary representation, signals the presence of the HairRightPart element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairLeftPartFlag	This field, which is only present in the binary representation, signals the presence of the HairLeftPart element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
HairPartBangsFlag	This field, which is only present in the binary representation, signals the presence of the HairPartBangs element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. ”1” means

		that the attribute shall be used. ”0” means that the attribute shall not be used.
	HairSize	The length of the hair (can be one of short, medium or long).
	HairStyle	The style of the hair as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the HairStyleCS defined in A.2.2.

Name	*Binary representation (8 bits)*	*Description*
afro	1	Afro hairstyle
bun	2	Bun hairstyle
combover	3	Combover hairstyle
crewcut	4	Crewcut hairstyle
mohawk	5	Mohawk hairstyle
odando	6	Odando hairstyle
pigtails	7	Pigtails hairstyle
pompadour	8	Pompadour hairstyle
ponytail	9	Ponytail hairstyle
	0,10-255	Reserved

	HairColor	The color type defined in ISO/IEC 23005-6 shall be used for hair colour.
	WhiteHair	Amount of white hair (%).
	RainbowColor	The color type defined in ISO/IEC 23005-6 shall be used for rainbow hair colour.
	BlondeHair	How much blond is the hair (%).
	RedHair	How much red is the hair (%).
	HairVolume	The volume of the complete hair (small, medium or big).
	HairFront	How much the hair goes toward front (short, medium or long).
	HairSides	The height of the sides of the hair (short, medium or long).
	HairBack	How long is the hair at the back (short, medium or long).
	BigHairFront	How high is the hair at the front of the skull (short, medium or long).
	BigHairTop	How high is the hair at the top of the skull (short, medium or long).
	BigHairBack	How high is the hair at the back of the skull (short, medium or long)
	FrontFringe	The length of the front fringe of the hair (short, medium or long).
	SideFringe	The length of the side fringe of the hair (short, medium or long).
	BackFringe	The length of the back fringe of the hair (short, medium or long).
	FullHairSides	The width of the hair (short, medium or long).
	HairSweep	How much the hair is turned towards the front (left, middle, right).
	ShearFront	How much the hair extends towards front (short, medium or long).
	ShearBack	How much the hair extends towards back (short, medium or long).
	TuperFront	The width of the hair at the front (short, medium or long).
	TuperBack	The width of the hair on the back (short, medium or long).
	Rumpledhair	How much the hair is rumpled (low, moderate or high).
	Pigtails	The length of the pigtails (short, medium or long).
	Ponytail	The length of the ponytail (short, medium or long).
	SpikedHair	The length of the spikes in the hair (short, medium or long).
	HairTilt	The vertical position of the hair from the top of the head (m).
	HairMiddlePart	How much the hair is parted at the middle front (low, high).
	HairRightPart	How much the hair is parted at the right side (low, high).

	HairLeftPart	How much the hair is parted at the left side (low, high).
	HairPartBangs	How much the hair is parted at the middle (low, high).
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the HairType.
	Extra	Describes any other descriptions of hair.
	hapticIDRef	Identifier that refers to the haptic properties of the hair.
Eyebrows	Set of descriptions for eyebrows.of the avatar.	
	Name	*Description*
	EyebrowsType	A type that describes avatar eyebrows.
	EyebrowSizeFlag	This field, which is only present in the binary representation, signals the presence of the EyebrowSize element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	EyebrowDensityFlag	This field, which is only present in the binary representation, signals the presence of the EyebrowDensity element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	EyebrowHeightFlag	This field, which is only present in the binary representation, signals the presence of the EyebrowHeight element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	EyebrowArcFlag	This field, which is only present in the binary representation, signals the presence of the EyebrowArc element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	EyebrowPointsFlag	This field, which is only present in the binary representation, signals the presence of the EyebrowPoints element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. ”1” means that the attribute shall be used. ”0” means that the attribute shall not be used.
	EyebrowSize	The length of the eyebrow (short, medium, long).
	EyebrowDensity	The density (low, moderate, high).
	EyebrowHeight	The vertical eyebrow position on the face (low, middle, high).
	EyebrowArc	The curvature of the eyebrow. It can be low (flat), middle or high (arced).
	EyebrowPoints	The direction of the eyebrows, towards up or down (down, middle, up).
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the EyebrowType.
	Extra	Describes any other descriptions of eyebrows.
	hapticIDRef	Identifier that refers to the haptic properties of the eyebrows.
FacialHair	Set of descriptions for facial hair of the avatar.	
	Name	*Description*
	FacialHairType	A type that describes avatar facial hair.
	FacialHairThicknessFlag	This field, which is only present in the binary representation, signals the presence of the FacialHairThickness element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
	FacialSideburnsFlag	This field, which is only present in the binary representation,

		signals the presence of the FacialSideburns element. "1" means that the element shall be used. "0" means that the element shall not be used.
	FacialMustacheFlag	This field, which is only present in the binary representation, signals the presence of the FacialMustache element. "1" means that the element shall be used. "0" means that the element shall not be used.
	FacialChinCurtainsFlag	This field, which is only present in the binary representation, signals the presence of the FacialChinCurtains element. "1" means that the element shall be used. "0" means that the element shall not be used.
	FacialSoulPatchFlag	This field, which is only present in the binary representation, signals the presence of the FacialSoulPatch element. "1" means that the element shall be used. "0" means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. "1" means that the element shall be used. "0" means that the element shall not be used.
	hapticIDRefFlag	This field, which is only present in the binary representation, signals the presence of the hapticIDRef attribute. "1" means that the attribute shall be used. "0" means that the attribute shall not be used.
	FacialHairThickness	The thick of the facial hair (low, middle, high).
	FacialSideBurns	The color type defined in ISO/IEC 23005-6 shall be used for the color of the facial side.
	FacialMoustache	The facial moustache (yes or no).
	FacialchinCurtains	Facial chin curtains (yes or no).
	FacialSoulPatch	Facial soul patch (yes or no).
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the FacialHairType.
	Extra	Describes any other descriptions of facial hair.
	hapticIDRef	Identifier that refers to the haptic properties of the facial hair.
BodyHair	Set of descriptions for body hair of the avatar.	
	Name	*Description*
	BodyHairType	A type that describes avatar body hair.
	HairColorFlag	This field, which is only present in the binary representation, signals the presence of the HairColor element. "1" means that the element shall be used. "0" means that the element shall not be used.
	HairThicknessFlag	This field, which is only present in the binary representation, signals the presence of the HairThickness element. "1" means that the element shall be used. "0" means that the element shall not be used.
	ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. "1" means that the element shall be used. "0" means that the element shall not be used.
	HairColor	The color type defined in ISO/IEC 23005-6 shall be used for avatar body hair.
	HairThickness	The thick of the body hair (low, middle, high).
	NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the BodyHairType.
	Extra	Describes any other descriptions of body hair.

Facial Calibration Points

Set of elements that are calibration points for the face feature control.

Name	*Description*
FacialCalibrationPointsType	A type that describes calibration points for face feature control.
SellionFlag	This field, which is only present in the binary representation, signals the presence of the Sellion element. "1" means that the element shall be used. "0" means that the element shall not be used.
RInfraorbitaleFlag	This field, which is only present in the binary representation, signals the presence of the RInfraorbitale element. "1" means that the element shall be used. "0" means that the element shall not be used.
LinfraorbitaleFlag	This field, which is only present in the binary representation, signals the presence of the Linfraorbitale element. "1" means that the element shall be used. "0" means that the element shall not be used.
SupramentonFlag	This field, which is only present in the binary representation, signals the presence of the Supramenton element. "1" means that the element shall be used. "0" means that the element shall not be used.
RtragionFlag	This field, which is only present in the binary representation, signals the presence of the Rtragion element. "1" means that the element shall be used. "0" means that the element shall not be used.
RgonionFlag	This field, which is only present in the binary representation, signals the presence of the Rgonion element. "1" means that the element shall be used. "0" means that the element shall not be used.
LtragionFlag	This field, which is only present in the binary representation, signals the presence of the Ltragion element. "1" means that the element shall be used. "0" means that the element shall not be used.
LgonionFlag	This field, which is only present in the binary representation, signals the presence of the Lgonion element. "1" means that the element shall be used. "0" means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType element. "1" means that the element shall be used. "0" means that the element shall not be used.
Sellion	3D position (meter), point 1 in the figure at the bottom.
RInfraorbitale	3D position (meter), point 2 in the figure at the bottom.
LInfraorbitale	3D position (meter), point 3 in the figure at the bottom.
Supramenton	3D position (meter), point 4 in the figure at the bottom.
RTragion	3D position (meter), point 5 in the figure at the bottom.
RGonion	3D position (meter), point 6 in the figure at the bottom.
LTragion	3D position (meter), point 7 in the figure at the bottom.
LGonion	3D position (meter), point 8 in the figure at the bottom.
NumExtra	This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the FacialCalibrationPointsType.
Extra	Describes any other descriptions of facial calibration points.

<table>
<tr><td></td><td>NOTE The calibration points are to be used for mapping a captured face feature points onto an arbitrary face of an avatar.</td></tr>
<tr><td>Physical Condition</td><td>This element contains a set of elements for describing the physical condition of the avatar.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>PhysicalConditionType</td><td>A type that describes the physical condition of the avatar.</td></tr>
<tr><td>BodyStrengthFlag</td><td>This field, which is only present in the binary representation, signals the presence of the BodyStrength element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td>BodyFlexibilityFlag</td><td>This field, which is only present in the binary representation, signals the presence of the BodyFlexibility element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td>ExtraFlag</td><td>This field, which is only present in the binary representation, signals the presence of the ExtraType element. ”1” means that the element shall be used. ”0” means that the element shall not be used.</td></tr>
<tr><td>BodyStrength</td><td>Avatar body strength [unlimited percentage (%)].</td></tr>
<tr><td>BodyFlexibility</td><td>Avatar body flexibility with descriptive scale of low, medium, and high.</td></tr>
<tr><td>NumExtra</td><td>This field, which is only present in the binary representation, specifies the number of ExtraType elements contained in the PhysicalConditionType.</td></tr>
<tr><td>Extra</td><td>Describes any other descriptions of physical condition.</td></tr>
</table></td></tr>
<tr><td>Clothes</td><td>A list of virtual clothes associated to the avatar. The type of this element is VirtualObjectType.</td></tr>
<tr><td>Shoes</td><td>A list of virtual shoes associated to the avatar. The type of this element is VirtualObjectType.</td></tr>
<tr><td>Accessories</td><td>A list of objects (ring, glasses, etc.) associated to the avatar. The type of this element is VirtualObjectType.</td></tr>
<tr><td>SkinMarks</td><td>A list of skin marks (birthmarks, scars, tattoos…, etc.) associated to the avatar. The type of this element is VirtualObjectType.</td></tr>
<tr><td>AppearanceResources</td><td>URL to file or streaming, containing the avatar visual representation. The avatar can be represented as 3D animated model, time-sequenced 3D model, 2D image, 2D video, 3D image, and 3D video, usually MP4 file.</td></tr>
<tr><td>Extra</td><td>Describes any other descriptions of avatar appearance.</td></tr>
</table>

5.2.4.4 Examples

This example shows the description of avatar appearance with the following semantics.

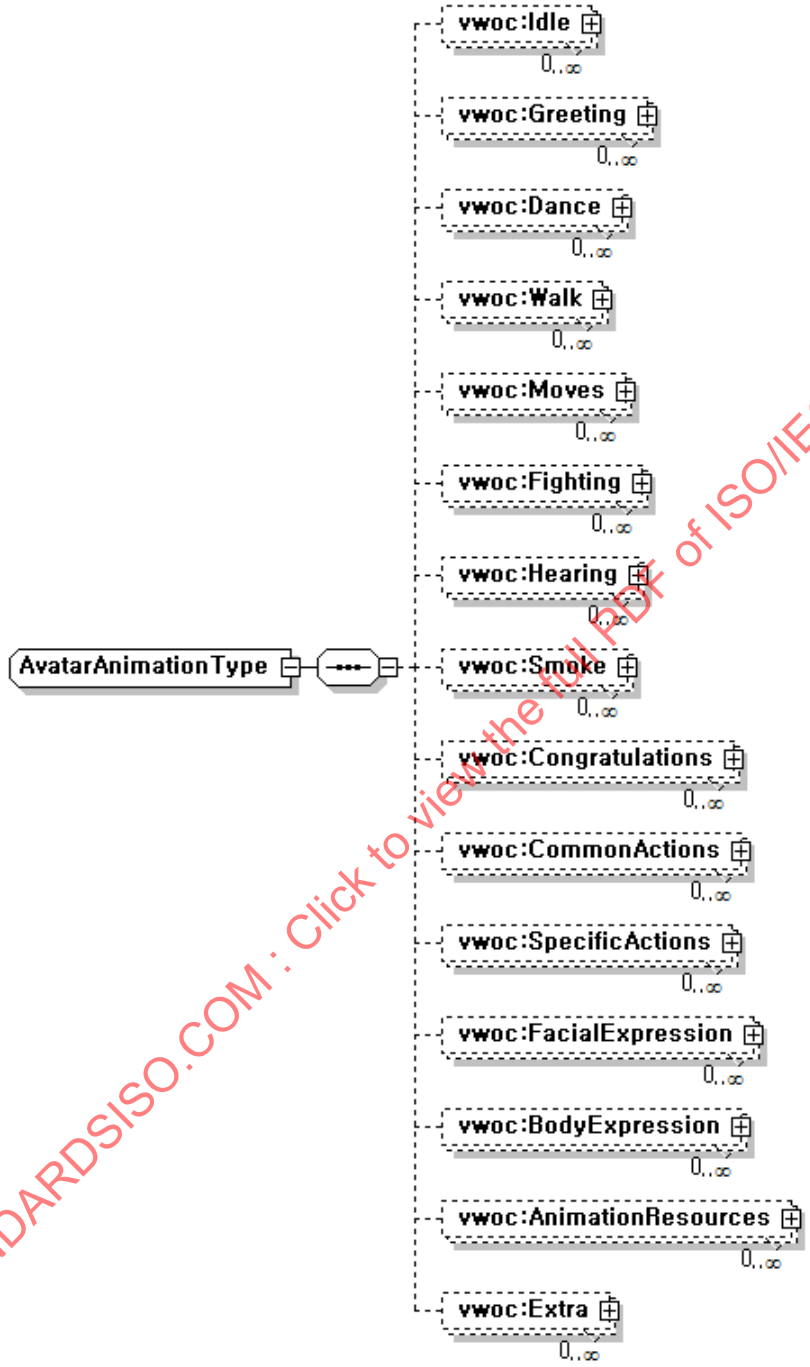
```
<vwoc:Appearance>
   <vwoc:Body>
      <vwoc:BodyHeight>5.2</vwoc:BodyHeight>
      <vwoc:BodyThickness>4.4</vwoc:BodyThickness>
      <vwoc:BodyFat>low</vwoc:BodyFat>
      <vwoc:TorsoMuscles>low</vwoc:TorsoMuscles>
      <vwoc:NeckThikness>2.1</vwoc:NeckThikness>
      <vwoc:NeckLength>1.8</vwoc:NeckLength>
      <vwoc:Package>small</vwoc:Package>
      <vwoc:SaddleBags>medium</vwoc:SaddleBags>
      <vwoc:KneeAngle>300</vwoc:KneeAngle>
      <vwoc:FootSize>3.1</vwoc:FootSize>
   </vwoc:Body>
   <vwoc:Head>
      <vwoc:HeadSize>small</vwoc:HeadSize>
      <vwoc:HeadStretch>1.1</vwoc:HeadStretch>
```

```
        <vwoc:HeadShape>square</vwoc:HeadShape>
        <vwoc:EggHead>true</vwoc:EggHead>
    </vwoc:Head>
    <vwoc:Eyes>
        <vwoc:EyeSize>1.1</vwoc:EyeSize>
    </vwoc:Eyes>
    <vwoc:Ears>
        <vwoc:EarSize>2.1</vwoc:EarSize>
    </vwoc:Ears>
    <vwoc:Nose>
        <vwoc:NoseSize>0.8</vwoc:NoseSize>
    </vwoc:Nose>
    <vwoc:FacialSkin>
        <vwoc:SkinRainbowColor>#FF8F69</vwoc:SkinRainbowColor>
    </vwoc:FacialSkin>
    <vwoc:ToeNails>
        <vwoc:NailPolish>true</vwoc:NailPolish>
        <vwoc:NailPolishColor>#CF8F69</vwoc:NailPolishColor>
    </vwoc:ToeNails>
    <vwoc:BodyLook>
        <vwoc:BodyDefinition>short</vwoc:BodyDefinition>
    </vwoc:BodyLook>
    <vwoc:Hair>
        <vwoc:HairSize>short</vwoc:HairSize>
        <vwoc:HairStyle>urn:mpeg:mpeg-v:01-VWOC-HairStyleCS-
NS:crewcut</vwoc:HairStyle>
    </vwoc:Hair>
    <vwoc:FacialCalibrationPoints>
        <vwoc:Sellion xsi:type="vwoc:Physical3DPointType" x="1.1" y="1.2"
z="1.2"/>
        <vwoc:RInfraorbitale xsi:type="vwoc:Physical3DPointType" x="1.1" y="1.2"
z="1.3"/>
    </vwoc:FacialCalibrationPoints>
    <vwoc:PhysicalCondition>
        <vwoc:BodyFlexibility>low</vwoc:BodyFlexibility>
    </vwoc:PhysicalCondition>
    <vwoc:Clothes id="vo_clothes_001">
        <vwoc:VirtualObjectComponents>
            <vwoc:VirtualObject xsi:type="vwoc:VirtualObjectType"
id="clothe_part_001">
                <vwoc:Appearance>id="virtualObject_001"</vwoc:Appearance>
            </vwoc:VirtualObject>
        </vwoc:VirtualObjectComponents>
    </vwoc:Clothes>
</vwoc:Appearance>
```

5.2.5 AvatarAnimationType

5.2.5.1 XML representation syntax

Diagram



Source

```
<complexType name="AvatarAnimationType">
    <sequence>
        <element name="Idle" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Greeting" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Dance" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Walk" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Moves" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="Fighting" type="vwoc:AnimationDescriptionType" minOccurs="0"
```

```
maxOccurs="unbounded"/>
            <element name="Hearing" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="Smoke" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="Congratulations" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="CommonActions" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="SpecificActions" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="FacialExpression" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="BodyExpression" type="vwoc:AnimationDescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <element name="AnimationResources" type="vwoc:AnimationResourcesDescriptionType"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="Extra" type="vwoc:ExtraType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
</complexType>
```

5.2.5.2 Binary representation syntax

AvatarAnimationType{	***Number of bits***	***Mnemonic***
IdleFlag	1	bslbf
GreetingFlag	1	bslbf
DanceFlag	1	bslbf
WalkFlag	1	bslbf
MovesFlag	1	bslbf
FightingFlag	1	bslbf
HearingFlag	1	bslbf
SmokeFlag	1	bslbf
CongratulationFlag	1	bslbf
CommonActionsFlag	1	bslbf
SpecificActionsFlag	1	bslbf
FacialExpressionFlag	1	bslbf
BodyExpressionFlag	1	bslbf
AnimationResourcesFlag	1	bslbf
ExtraFlag	1	bslbf
if(IdleFlag){		
NumIdle		vluimsbf5

for(k=0; k< NumIdle; k++){		
Idle[k]		AnimationDescriptionType
}		
}		
if(GreetingFlag){		
NumGreeting		vluimsbf5
for(k=0; k< NumGreeting; k++){		
Greeting[k]		AnimationDescriptionType
}		
}		
if(DanceFlag){		
NumDance		vluimsbf5
for(k=0; k< NumDance; k++){		
Dance[k]		AnimationDescriptionType
}		
}		
if(WalkFlag){		
NumWalk		vluimsbf5
for(k=0; k< NumWalk; k++){		
Walk[k]		AnimationDescriptionType
}		
}		
if(MovesFlag){		
NumMoves		vluimsbf5
for(k=0; k< NumMoves; k++){		
Moves[k]		AnimationDescriptionType

}		
}		
if(FightingFlag){		
NumFighting		vluimsbf5
for(k=0; k< NumFighting; k++){		
Fighting[k]		AnimationDescriptionType
}		
}		
if(HearingFlag){		
NumHearing		vluimsbf5
for(k=0; k< NumHearing; k++){		
Hearing[k]		AnimationDescriptionType
}		
}		
if(SmokeFlag){		
NumSmoke		vluimsbf5
for(k=0; k< NumSmoke; k++){		
Smoke[k]		AnimationDescriptionType
}		
}		
if(CongratulationsFlag){		
NumCongratulations		vluimsbf5
for(k=0; k< NumCongratulations; k++){		
Congratulations[k]		AnimationDescriptionType
}		
}		

if(CommonActionsFlag){		
NumCommonActions		vluimsbf5
for(k=0; k< NumCommonActions; k++){		
CommonActions[k]		AnimationDescriptionType
}		
}		
if(SpecificActionsFlag){		
NumSpecificActions		vluimsbf5
for(k=0; k< NumSpecificActions; k++){		
SpecificActions[k]		AnimationDescriptionType
}		
}		
if(FacialExpressionFlag){		
NumFacialExpression		vluimsbf5
for(k=0; k< NumFacialExpression; k++){		
FacialExpression[k]		AnimationDescriptionType
}		
}		
if(BodyExpressionFlag){		
NumBodyExpression		vluimsbf5
for(k=0; k< NumBodyExpression; k++){		
BodyExpression[k]		AnimationDescriptionType
}		
}		
if(AnimationResourcesFlag){		

NumAnimationResources		vluimsbf5
for(k=0; k< NumAnimationResources; k++){		
AnimationResources[k]		AnimationResourcesDescriptionType
}		
}		
if(ExtraFlag){		
NumExtra		vluimsbf5
for(k=0; k< NumExtra; k++){		
Extra[k]		ExtraType
}		
}		
}		

5.2.5.3 Semantics

Name	*Description*
`AvatarAnimationType`	A type that contains the description of a set of animation sequences that the avatar is able to perform and may refer to several medias containing the exact (geometric transformations) animation parameters.
`IdleFlag`	This field, which is only present in the binary representation, signals the presence of the `Idle` elements. "1" means that the element shall be used. "0" means that the element shall not be used.
`GreetingFlag`	This field, which is only present in the binary representation, signals the presence of the `Greeting` elements. "1" means that the element shall be used. "0" means that the element shall not be used.
`DanceFlag`	This field, which is only present in the binary representation, signals the presence of the `Dance` elements. "1" means that the element shall be used. "0" means that the element shall not be used.
`WalkFlag`	This field, which is only present in the binary representation, signals the presence of the `Walk` elements. "1" means that the element shall be used. "0" means that the element shall not be used.
`MovesFlag`	This field, which is only present in the binary representation, signals the presence of the `Moves` elements. "1" means that the element shall be used. "0" means that the element shall not be used.
`FightingFlag`	This field, which is only present in the binary representation, signals the presence of the `Fighting` elements. "1" means that the element shall be used. "0" means that the element shall not be used.
`HearingFlag`	This field, which is only present in the binary representation, signals the presence of the `Hearing` elements. "1" means that the element shall be used. "0" means that the element shall not be used.
`SmokeFlag`	This field, which is only present in the binary representation, signals the presence of the `Smoke` elements. "1" means that the element shall be used. "0" means that the element shall not be used.
`CongratulationsFla`	This field, which is only present in the binary representation, signals the presence

g	of the Congratulations elements. ”1” means that the element shall be used. ”0” means that the element shall not be used.
CommonActionsFlag	This field, which is only present in the binary representation, signals the presence of the CommonActions elements. ”1” means that the element shall be used. ”0” means that the element shall not be used.
SpecificActionsFlag	This field, which is only present in the binary representation, signals the presence of the SpecificActions elements. ”1” means that the element shall be used. ”0” means that the element shall not be used.
FacialExpressionFlag	This field, which is only present in the binary representation, signals the presence of the FacialExpression elements. ”1” means that the element shall be used. ”0” means that the element shall not be used.
BodyExpressionFlag	This field, which is only present in the binary representation, signals the presence of the BodyExpression elements. ”1” means that the element shall be used. ”0” means that the element shall not be used.
AnimationResourcesFlag	This field, which is only present in the binary representation, signals the presence of the AnimationResources elements. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ExtraFlag	This field, which is only present in the binary representation, signals the presence of the ExtraType elements. ”1” means that the element shall be used. ”0” means that the element shall not be used.
NumIdle	This field, which is only present in the binary representation, signals the number of the Idle elements.
NumGreeting	This field, which is only present in the binary representation, signals the number of the Greeting elements.
NumDance	This field, which is only present in the binary representation, signals the number of the Dance elements.
NumWalk	This field, which is only present in the binary representation, signals the number of the Walk elements.
NumMoves	This field, which is only present in the binary representation, signals the number of the Moves elements.
NumFighting	This field, which is only present in the binary representation, signals the number of the Fighting elements.
NumHearing	This field, which is only present in the binary representation, signals the number of the Hearing elements.
NumSmoke	This field, which is only present in the binary representation, signals the number of the Smoke elements.
NumCongratulations	This field, which is only present in the binary representation, signals the number of the Congratulations elements.
NumCommonActions	This field, which is only present in the binary representation, signals the number of the CommonActions elements.
NumSpecificActions	This field, which is only present in the binary representation, signals the number of the SpecificActions elements.
NumFacialExpression	This field, which is only present in the binary representation, signals the number of the FacialExpression elements.
NumBodyExpression	This field, which is only present in the binary representation, signals the number of the BodyExpression elements.
NumAnimationResources	This field, which is only present in the binary representation, signals the number of the AnimationResources elements.
Idle	Describes an idle type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the IdleAnimationCS defined in A.2.3.1.

Name	*Binary representation (4 bits)*	*Description*
defaultIdle	1	Default idle
restPose	2	Rest pose

<table>
<tr><td></td><td>
<table>
<tr><td>breathe</td><td>3</td><td>Breathe</td></tr>
<tr><td>bodyNoise</td><td>4</td><td>Body noise</td></tr>
<tr><td></td><td>0,5-15</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>Greeting</td><td>Describes a greeting type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the GreetingAnimationCS defined in A.2.3.2.
<table>
<tr><td>Name</td><td>Binary representation (4 bits)</td><td>Description</td></tr>
<tr><td>salute</td><td>1</td><td>Salute</td></tr>
<tr><td>cheer</td><td>2</td><td>Cheer</td></tr>
<tr><td>greet</td><td>3</td><td>Greet</td></tr>
<tr><td>wave</td><td>4</td><td>Wave</td></tr>
<tr><td>hello</td><td>5</td><td>Hello</td></tr>
<tr><td>bow</td><td>6</td><td>Bow</td></tr>
<tr><td>courtBow</td><td>7</td><td>Court bow</td></tr>
<tr><td>flourish</td><td>8</td><td>Flourish</td></tr>
<tr><td></td><td>0,8-15</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Dance</td><td>Describes a dance type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the DanceAnimationCS defined in A.2.3.3.
<table>
<tr><th>Name</th><th>Binary representation (5 bits)</th><th>Description</th></tr>
<tr><td>bodyPopDance</td><td>1</td><td>Body pop dance</td></tr>
<tr><td>breakDance</td><td>2</td><td>Break dance</td></tr>
<tr><td>cabbagePatchDance</td><td>3</td><td>Cabbage patch dance</td></tr>
<tr><td>casualDance</td><td>4</td><td>Casual dance</td></tr>
<tr><td>dance</td><td>5</td><td>Dance</td></tr>
<tr><td>raveDance</td><td>6</td><td>Rave dance</td></tr>
<tr><td>robotDance</td><td>7</td><td>Robot dance</td></tr>
<tr><td>rockDance</td><td>8</td><td>Rock dance</td></tr>
<tr><td>rockRollDance</td><td>9</td><td>Rock and roll dance</td></tr>
<tr><td>runningManDance</td><td>10</td><td>Running man dance</td></tr>
<tr><td>salsaDance</td><td>11</td><td>Salsa dance</td></tr>
<tr><td></td><td>0,12-31</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>Walk</td><td>Describes a Walk type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the WalkAnimationCS defined in A.2.3.4.
<table>
<tr><th>Name</th><th>Binary representation (5 bits)</th><th>Description</th></tr>
<tr><td>slowWalk</td><td>1</td><td>Slow walk</td></tr>
<tr><td>defaultWalk</td><td>2</td><td>Default walk</td></tr>
<tr><td>fastWalk</td><td>3</td><td>Fast walk</td></tr>
<tr><td>slowRun</td><td>4</td><td>Slow run</td></tr>
<tr><td>defaultRun</td><td>5</td><td>Default run</td></tr>
<tr><td>fastRun</td><td>6</td><td>Fast run</td></tr>
<tr><td>crouch</td><td>7</td><td>Crouch</td></tr>
<tr><td>crouchWalk</td><td>8</td><td>Crouch walk</td></tr>
<tr><td></td><td>0,9-31</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>Moves</td><td>Describes a moves type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the MovesAnimationCS defined in A.2.3.5.</td></tr>
</table>

Name	*Binary representation (5 bits)*	*Description*
moveDown	1	Move down
moveLeft	2	Move left
moveRight	3	Move right
moveUp	4	Move up
pointMe	5	Point me
pointYou	6	Point you
turn180	7	Turn 180
turnBack180	8	Turn back 180
turnLeft	9	Turn left
turnRight	10	Turn right
turn360	11	Turn 360
turnBack360	12	Turn back 360
freeDirection	13	Free direction
	0,14-31	Reserved

`Fighting`

Describes a `Figting` type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the `FightingAnimationCS` defined in A.2.3.6.

Name	*Binary representation (7 bits)*	*Description*
aim	1	Aim
aimLeft	2	Aim left
aimRight	3	Aim right
aimBow	4	Aim bow
aimLeftBow	5	Aim left bow
aimRightBow	6	Aim right bow
aimLeftRifle	7	Aim left rifle
aimRightRifle	8	Aim right rifle
aimBazooka	9	Aim bazooka
aimLeftBazooka	10	Aim left bazooka
aimRightBazooka	11	Aim right bazooka

aimHandgun	12	Aim handgun
aimLeftHandgun	13	Aim left handgun
aimRightHandgun	14	Aim right handgun
holdWeapon	15	Hold weapon
holdWeaponLeft	16	Hold weapon left
holdWeaponRight	17	Hold weapon right
holdBow	18	Hold bow
holdBowLeft	19	Hold bow left
holdBowRight	20	Hold bow right
holdRifle	21	Hold rifle
holdRifleLeft	22	Hold rifle left
holdRifleRight	23	Hold rifle right
holdBazooka	24	Hold bazooka
holdBazookaLeft	25	Hold bazooka left
holdBazookaRight	26	Hold bazooka right
holdHandgun	27	Hold handgun
holdHandgunLeft	28	Hold handgun left
holdHandgunRight	29	Hold handgun right
holdWeaponThrow	30	Hold weapon throw
holdWeaponThrowReft	31	Hold weapon throw left
holdWeaponThrowRight	32	Hold weapon throw right
shoot	33	shoot
shootLeft	34	shoot left
shootRight	35	shoot right
shootBow	36	Shoot bow
shootBowLeft	37	Shoot bow left
shootBowRight	38	Shoot bow right
shootRifle	39	Shoot rifle

	shootRifleLeft	40	Shoot rifle left
	shootRifleRight	41	Shoot rifle right
	shootBazooka	42	Shoot bazooka
	shootBazookaLeft	43	Shoot bazooka left
	shootBazookaRight	44	Shoot bazooka right
	shootHandgun	45	Shoot handgun
	shootHandgunLeft	46	Shoot handgun left
	shootHandgunRight	47	Shoot handgun right
	strike	48	Strike
	strikeSword	49	Strike sword
	strikeSwordLeft	50	Strike sword left
	strikeSwordRight	51	Strike sword right
	punch	52	Punch
	punchLeft	53	Punch left
	punchRight	54	Punch right
	throwing	55	Throwing
	throwWeaponLeft	56	Throw weapon left
	throwWeaponRight	57	Throw weapon right
		0, 58-127	Reserved

`Hearing`

Describes a `Hearing` type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the `HearingAnimationCS` defined in A.2.3.7.

Name	*Binary representation (5 bits)*	*Description*
startHearing	1	Start hearing
stopHearing	2	Stop hearing
earsExtend	3	Ears extend
turnsHeadLeft	4	Turns head left
turnsHeadRight	5	Turns head right
holdsUpHand	6	Holds up hand
tiltsHeadRight	7	Tilts head right

	tiltsHeadLeft	8	Tilts head left
	cocksHeadLeft	9	Cocks head left
	defaultHear	10	Default hear
		0,11-31	Reserved

`Smoke`

Describes a `Smoke` type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the `SmokeAnimationCS` defined in A.2.3.8.

Name	*Binary representation (4 bits)*	*Description*
smokeIdle	1	Smoke idle
smokeInhale	2	Smoke inhale
smokeThrowDown	3	Smoke throw down
	0, 4-15	Reserved

`Congratulations`

Describes a `Congratulations` type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the `CongratulationsAnimationCS` defined in A.2.3.9.

Name	*Binary representation (4 bits)*	*Description*
applaude	1	Applaud
clap	2	Clap
	0, 3-15	Reserved

`CommonActions`

Describes a `CommonActions` type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the `CommonActionsAnimationCS` defined in A.2.3.10.

Name	*Binary representation (7 bits)*	*Description*
appear	1	Appear
away	2	Away
blowKiss	3	Blow kiss
brush	4	Brush
busy	5	Busy
crazy	6	Crazy
dead	7	Dead
disappear	8	Disappear
drink	9	Drink

eat	10	Eat
explain	11	Explain
fallDown	12	Fall down
flip	13	Flip
fly	14	Fly
gag	15	Gag
getAttention	16	Get attention
impatient	17	Impatient
jump	18	Jump
kick	19	Kick
land	20	Land
prejump	21	Prejump
puke	22	Puke
read	23	Read
sit	24	Sit
sleep	25	Sleep
stand	26	Stand
standUp	27	Stand up
stretch	28	Stretch
stride	29	Stride
suggest	30	Suggest
surf	31	Surf
talk	32	Talk
think	33	Think
type	34	Type
whisper	35	Whisper
whistle	36	Whistle
write	37	Write
yawn	38	Yawn
yeah	39	Yeah

	yoga	40	Yoga
		0, 41-127	Reserved

`SpecificActions`

Describes a `SpecificActions` type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the `SpecificActionsAnimationCS` defined in A.2.3.11.

Name	*Binary representation (8 bits)*	*Description*
airGuitar	1	air guitar
angryFingerWag	2	angry_fingerwag
angryTantrum	3	angry_tantrum
backFlip	4	back flip
beckOn	5	beck on
bigYawn	6	big yawn
boo	7	boo
burp	8	burp
candleStick	9	candle Stick
comeAgain	10	come again
decline	11	decline
dismissive	12	Dismissive
dontRecognize	13	don't recognize
fartArm	14	fart arm
fistPump	15	fist pump
flySlow	16	fly slow
guns	17	guns
ha	18	ha
hide	19	hide
hmmm	20	hmmm
hover	21	hover
hoverDown	22	hover down
hoverUp	23	hover up
huh	24	Huh

jumpForJoy	25	jump for joy
kickRoundHouse	26	kick roundhouse
kissMyButt	27	kiss my butt
laughtShort	28	laught short
lol	29	lol
loser	30	loser
motorcycleSit	31	motorcycle sit
muscleBeach	32	muscle beach
noWay	33	no way
noHead	34	no head
noUnhappy	35	no unhappy
nod	36	nod
nope	37	Nope
nyanya	38	nyanya
okay	39	okay
oooh	40	oooh
peace	41	peace
point	42	point
pose	43	pose
punchOneTwo	44	punch one two
rpsCountDown	45	rps countdown
rpsPaper	46	rps paper
rpsRock	47	rps rock
rpsScissors	48	rps scissors
score	49	score
shakeFists	50	shake fists
show	51	show
sitGeneric	52	sit generic
sitGround	53	sit ground
sitGroundConstrain	54	sit ground constrained

	ed		
	sitToStand	55	sit to stand
	slowFly	56	slow fly
	snapshot	57	snapshot
	softLand	58	soft land
	spin	59	spin
	tantrum	60	tantrum
	thumbsDown	61	thumbs down
	thumbsUp	62	thumbs up
	tongue	63	Tongue
	tryonShirt	64	tryon shirt
	uncertain	65	uncertain
	wassamatta	66	wassamatta
	what	67	what
	yay	68	yay
	yesHappy	69	yes happy
	yesHead	70	yes head
		0, 71-255	Reserved

`FacialExpression`

Describes a `FacialExpression` type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the `FacialExpressionAnimationCS` defined in A.2.3.12.

Name	*Binary representation (8 bits)*	*Description*
affection	1	Affected face
afraid	2	Afraid face
agree	3	Agree face
amusement	4	Amused face
angry	5	Angry face
annoyance	6	Annoyance face
anxiety	7	Anxiety face
bigSmile	8	Big smile face
blink	9	Blink face

bored	10	Bored face	
calm	11	Calm face	
concentrate	12	Concentrate face	
confused	13	Confused face	
contempt	14	Contempt face	
content	15	Content face	
courage	16	Courage face	
cry	17	Cry face	
dazed	18	Dazed face	
defaultEmotion	19	Default emotion face	
delight	20	Delight face	
despair	21	Despair face	
disagree	22	Disagree face	
disappointment	23	Disappointment face	
disdain	24	Disdain face	
disgusted	25	Disgusted face	
doubt	26	Doubt face	
elation	27	Elation face	
embarrassed	28	Embarrassed face	
empathy	29	Empathy face	
envy	30	Envy face	
excitement	31	Excitement face	
fear	32	Fear face	
friendliness	33	Friendliness face	
frown	34	Frown face	
frustration	35	Frustration face	
grin	36	Grin face	
guilt	37	Guilt face	
happy	38	Happy face	
helplessness	39	Helpless face	

	hope	40	Hoping face	
	hurt	41	Hurt face	
	interest	42	Interested face	
	irritation	43	Irritated face	
	joy	44	Joy face	
	kiss	45	Kiss face	
	laugh	46	Laughing face	
	lookDown	47	Look down face	
	lookDownBlink	48	Look down blink face	
	lookDownLeft	49	Look down left face	
	lookDownLeftBlink	50	Look down left blink face	
	lookDownLeftReturn	51	Look down left return face	
	lookDownReturn	52	Look down return face	
	lookDownRight	53	Look down right face	
	lookDownRightBlink	54	Look down right blink face	
	lookDownRightReturn	55	Look down right return face	
	lookLeft	56	Look left face	
	lookLeftBlink	57	Look left blink face	
	lookLeftReturn	58	Look left return face	
	lookRight	59	Look right face	
	lookRightBlink	60	Look right blink face	
	lookRightReturn	61	Look right return face	
	lookUp	62	Look up face	
	lookUpBlink	63	Look up blink face	
	lookUpLeft	64	Look up left face	

lookUpLeftBlink	65	Look up left blink face	
lookUpLeftReturn	66	Look up left return face	
lookUpReturn	67	Look up return face	
lookUpRight	68	Look up right face	
lookUpRightBlink	69	Look up right blink face	
lookUpRightReturn	70	Look up left return face	
love	71	Love face	
mad	72	Mad face	
neutral	73	Neutral face	
openMouth	74	Open mouth face	
pleasure	75	pleased face	
politeness	76	polite face	
powerlessness	77	Powerlessness face	
pride	78	Pride face	
pucker	79	Puckering	
relaxed	80	Relaxed face	
relieved	81	Relieved face	
repulsed	82	Repulsed face	
sad	83	Sad face	
satisfaction	84	Satisfied face	
scream	85	Screaming	
serene	86	Serene face	
shame	87	Shame face	
shock	88	Shocked face	
shrug	89	Shrug face	
sigh	90	Sigh face	
smile	91	Smiling face	
stress	92	Stressed face	

	surprise	93	Surprised face	
	tension	94	Tension face	
	tongueOut	95	Tongue out face	
	toothSmile	96	Tooth smile face	
	tired	97	Tired face	
	trust	98	Trust face	
	wink	99	Wink face	
	worry	100	Worried face	
	gestureRight	101	Gesture right face	
	gestureLeft	102	Gesture left face	
	gestureUp	103	Gesture up face	
	gestureDown	104	Gesture down face	
		0, 105-255	Reserved	

`BodyExpression`

Describes a `BodyExpression` type of animations as a reference to a classification scheme (CS) term. A CS that may be used for this purpose is the `BodyExpressionAnimationCS` defined in A.2.3.13.

Name	*Binary representation (8 bits)*	*Description*	
affection	1	affected pose	
afraid	2	afraid pose	
agree	3	agree pose	
amusement	4	amuse pose	
angry	5	angry pose	
annoyance	6	annoyance pose	
anxiety	7	anxiety pose	
bored	8	bored pose	
calm	9	calm pose	
concentrate	10	concentrate pose	
confused	11	confused pose	
contempt	12	contempt pose	
content	13	content pose	

	courage	14	courage pose	
	cry	15	cry pose	
	dazed	16	dazed pose	
	delight	17	delight pose	
	despair	18	despair pose	
	disagree	19	disagree pose	
	disappointment	20	disappointed pose	
	disdain	21	disdain pose	
	disgusted	22	disgusted pose	
	doubt	23	doubt pose	
	elation	24	elation pose	
	embarrassed	25	embarrassed pose	
	empathy	26	empathy pose	
	envy	27	envy pose	
	excitement	28	excitement pose	
	fear	29	fear pose	
	friendliness	30	friendliness pose	
	frown	31	frown pose	
	frustration	32	frustrated pose	
	grin	33	grin pose	
	guilt	34	guilt pose	
	happy	35	happy pose	
	helplessness	36	helplessness pose	
	hope	37	hoping pose	
	hurt	38	hurt pose	
	interest	39	interested pose	
	irritation	40	irritated pose	
	joy	41	joy pose	
	laugh	42	laughing pose	
	love	43	love pose	

	mad	44	mad pose	
	neutral	45	neutral pose	
	pleasure	46	pleasure pose	
	politeness	47	politeness pose	
	powerlessness	48	powerlessness pose	
	pride	49	pride pose	
	pucker	50	puckering	
	relaxed	51	relaxed pose	
	relieved	52	relieved pose	
	repulsed	53	repulsed pose	
	sad	54	sad pose	
	satisfied	55	satisfied pose	
	scream	56	screaming	
	serene	57	serene pose	
	shame	58	shame pose	
	shock	59	shocked pose	
	shrug	60	shrug pose	
	sigh	61	sigh pose	
	smile	62	smiling pose	
	stress	63	stressed pose	
	surprise	64	surprised pose	
	tension	65	tension pose	
	tired	66	tired pose	
	worry	67	worried pose	
		0, 68-255	Reserved	

`AnimationResources`	Element that contains a link to animation file.
`NumExtra`	This field, which is only present in the binary representation, specifies the number of `ExtraType` elements contained in the `AnimationType`.
`Extra`	Describes any other categories of animations.

5.2.5.4 Examples

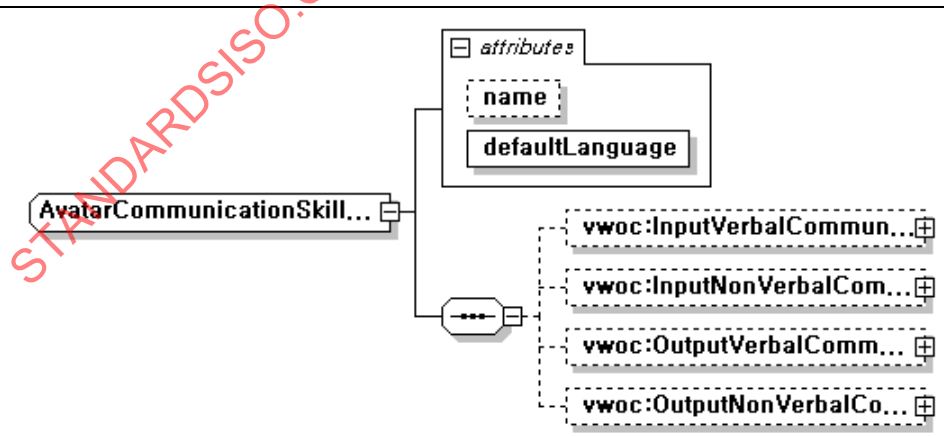
This example shows the description of avatar animation information with the following semantics. Among all animations, idle at default, saluting greeting, bow, dance, and salsa dance are given. The animation resources are saved at "http://avatarAnimationdb.com/default_idle.bvh", "http://avatarAnimationdb.com/salutes.bvh", "http://avatarAnimationdb.com/bowing.bvh", "http://avatarAnimationdb.com/dancing.bvh", and "http://avatarAnimationdb.com/salsa.bvh".

```
<vwoc:Animation>
    <vwoc:Idle>
        <vwoc:Name>urn:mpeg:mpeg-v:01-VWOC-IdleAnimationCS-
NS:defaultIdle</vwoc:Name>
        <vwoc:Uri>http://avatarAnimationdb.com/default_idle.bvh</vwoc:Uri>
    </vwoc:Idle>
    <vwoc:Greeting>
        <vwoc:Name>urn:mpeg:mpeg-v:01-VWOC-GreetingAnimationCS-
NS:salute</vwoc:Name>
        <vwoc:Uri>http://avatarAnimationdb.com/salutes.bvh</vwoc:Uri>
    </vwoc:Greeting>
    <vwoc:Greeting>
        <vwoc:Name>urn:mpeg:mpeg-v:01-VWOC-GreetingAnimationCS-NS:bow</vwoc:Name>
        <vwoc:Uri>http://avatarAnimationdb.com/bowing.bvh</vwoc:Uri>
    </vwoc:Greeting>
    <vwoc:Dance>
        <vwoc:Name>urn:mpeg:mpeg-v:01-VWOC-DanceAnimationCS-NS:dance</vwoc:Name>
        <vwoc:Uri>http://avatarAnimationdb.com/dancing.bvh</vwoc:Uri>
    </vwoc:Dance>
    <vwoc:Dance>
        <vwoc:Name>urn:mpeg:mpeg-v:01-VWOC-DanceAnimationCS-
NS:salsaDance</vwoc:Name>
        <vwoc:Uri>http://avatarAnimationdb.com/salsa.bvh</vwoc:Uri>
    </vwoc:Dance>
</vwoc:Animation>
```

5.2.6 AvatarCommunicationSkillsType

This element defines the communication skills[3] of the avatar in relation to other avatars.

5.2.6.1 XML representation syntax

Diagram	
Source	<complexType name="AvatarCommunicationSkillsType"> <sequence> <element name="InputVerbalCommunication" type="vwoc:VerbalCommunicationType" minOccurs="0"/> <element name="InputNonVerbalCommunication" type="vwoc:NonVerbalCommunicationType" minOccurs="0"/> <element name="OutputVerbalCommunication" type="vwoc:VerbalCommunicationType" minOccurs="0"/> <element name="OutputNonVerbalCommunication" type="vwoc:NonVerbalCommunicationType"


```
minOccurs="0"/>
        </sequence>
        <attribute name="name" type="string"/>
        <attribute name="defaultLanguage" type="language" use="required"/>
</complexType>
```

5.2.6.2 Binary representation syntax

AvatarCommunicationSkillsType{	***Number of bits***	***Mnemonic***
InputVerbalCommunicationFlag	1	bslbf
InputNonVerbalCommunicationFlag	1	bslbf
OutputVerbalCommunicationFlag	1	bslbf
OutputNonVerbalCommaunicationFlag	1	bslbf
NameFlag	1	bslbf
if(InputVerbalCommunicationFlag){		
InputVerbalCommunication		VerbalCommunicationType
}		
if(InputNonVerbalCommunicationFlag){		
InputNonVerbalCommunication		NonVerbalCommunicationType
}		
if(OutputVerbalCommunicationFlag){		
OutputVerbalCommunication		VerbalCommunicationType
}		
if(OutputNonVerbalCommaunicationFlag){		
OutputNonVerbalCommaunication		NonVerbalCommaunicationType
}		
if(NameFlag){		
name	See ISO 10646	UTF-8
}		

DefaultLanguage	See ISO 10646	UTF-8
}		

5.2.6.3 Semantics

The objective of the type is that the virtual world and the rest of avatars can adapt their inputs and outputs to these preferences (having a balance with their own preferences too). All inputs and outputs will be individually adapted for each avatar.

The communication preferences are defined by means of two input and two output channels that guarantee multimodality. They are the verbal and non-verbal recognition as input, and the verbal and non-verbal performance as output. These channels can be specified as either enabled or disabled. All channels enabled imply that an avatar is able to speak, to perform gestures and to recognize speak and gestures.

In verbal performance and verbal recognition channels the preference for using the channel either via text or via voice can be specified.

The non-verbal and non-verbal recognition channels specify the types of gesturing: "Nonverbal language", "sign language" and "cued speech communication"[2].

All the features dependent on the language (speaking via text or voice, speaking recognition via text or voice, and sign/cued language use/recognition) use a language attribute for defining the concrete language skills.

Name	*Definition*
`AvatarCommunicationSkillsType`	A type that contains a set of descriptors providing information on the different modalities an avatar is able to communicate.
`InputVerbalCommunicationFlag`	This field, which is only present in the binary representation, signals the presence of the `InputVerbalCommunication` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`InputNonVerbalCommunicationFlag`	This field, which is only present in the binary representation, signals the presence of the `InputNonVerbalCommunication` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`OutputVerbalCommunicationFlag`	This field, which is only present in the binary representation, signals the presence of the `OutputVerbalCommunication` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`OutputNonVerbalCommaunicationFlag`	This field, which is only present in the binary representation, signals the presence of the `OutputNonVerbalCommaunication` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`NameFlag`	This field, which is only present in the binary representation, signals the presence of the `Name` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`VerbalCommunicationType`	Defines the verbal (voice and text) communication skills of the avatar.
`NonVerbalCommunicationType`	Defines the non-verbal (body gesture) communication skills of the avatar.
`name`	A user defined chain of characters used for addressing the `CommunicationType` element.
`defaultLanguage`	The native language of the avatar (ex. en for English, es for Spanish. The language shall be written according to the

	ISO 639 which describes the set of international standards that lists short codes for language names.) Note: defaultLanguage attribute specifies the avatar's preferred language for all the communication channels (it will be generally its native language). For each communication channel, other languages that override this preference can be specified.

NOTE Additional information about ISO 639 can be found in Annex D.

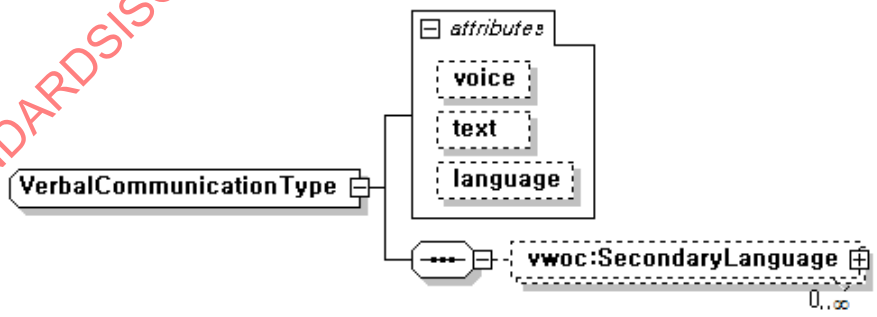
5.2.6.4 Examples

This example shows the description of avatar communication skills with the following semantics. The communication skills have a name of "Korean" which has the default language as "Korean". The preference of the primary input verbal communication is "Korean" as a language preferred for both voice and text. In addition, the secondary input verbal communication is English as a language with the preference of voice. As for the input non-verbal communication, "nod" is chosen for the complementary gesture. The preference of the primary output verbal communication is "Korean" as a language preferred for both voice and text. The secondary output verbal communication is "English" as a language with the preference of "voice". As for the output non-verbal communication, "nod" is chosen for the complementary gesture.

```
<vwoc:CommunicationSkills defaultLanguage="Korean" name="Korean">
   <vwoc:InputVerbalCommunication voice="preferred" text="preferred"
language="Korean">
      <vwoc:SecondaryLanguage preference="voice" name="English"/>
   </vwoc:InputVerbalCommunication>
   <vwoc:InputNonVerbalCommunication complementaryGesture="nod"/>
   <vwoc:OutputVerbalCommunication voice="preferred" text="preferred"
language="Korean">
      <vwoc:SecondaryLanguage preference="voice" name="English"/>
   </vwoc:OutputVerbalCommunication>
   <vwoc:OutputNonVerbalCommunication complementaryGesture="nod"/>
</vwoc:CommunicationSkills>
```

5.2.6.5 VerbalCommunicationType

5.2.6.5.1 XML representation syntax

Diagram	
Source	<complexType name="VerbalCommunicationType"> <sequence> <element name="SecondaryLanguage" type="vwoc:LanguageType" minOccurs="0" maxOccurs="unbounded"/> </sequence> <attribute name="voice" type="vwoc:communicationPreferenceLevelType"/> <attribute name="text" type="vwoc:communicationPreferenceLevelType"/> <attribute name="language" type="language"/> </complexType>

5.2.6.5.2 Binary representation syntax

VerbalCommunicationType{	***Number of bits***	***Mnemonic***
VoiceFlag	1	bslbf
TextFlag	1	bslbf
LanguageFlag	1	bslbf
SecondaryLanguageFlag	1	bslbf
if(VoiceFlag){		
voice		communicationPreferenceLevelType
}		
if(TextFlag){		
text		communicationPreferenceLevelType
}		
if(LanguageFlag){		
language	See ISO 10646	UTF-8
}		
if(SecondaryLanguageFlag){		
NumSecondaryLanguage		
for(k=0; k<NumSecondaryLanguage; k++) {		vluimsbf5
SecondaryLanguage[k]		LanguageType
}		
}		
}		

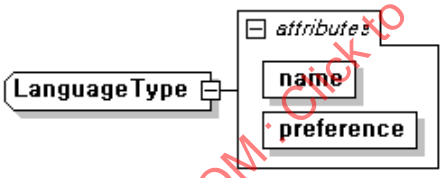
5.2.6.5.3 Semantics

Name	*Definition*
`VerbalCommunicationType`	Specifies the avatar's verbal communication skills. Voice and text can be defined as enabled, disabled or preferred in order to specify what the preferred verbal mode is and the availability of the other.
`VoiceFlag`	This field, which is only present in the binary representation, signals the presence of the `Voice` element. "1" means that the element shall be

	used. ”0” means that the element shall not be used.
TextFlag	This field, which is only present in the binary representation, signals the presence of the Text element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
PreferredLanguageFlag	This field, which is only present in the binary representation, signals the presence of the PreferredLanguage element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
LanguageFlag	This field, which is only present in the binary representation, signals the presence of the Language element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
PreferredLanguageLength	This field, which is only present in the binary representation, specifies the length of the following PreferredLanguage element.
SecondaryLanguage	Defines the preferred language for verbal communication according to the ISO 639 which describes the set of International Standards that lists short codes for language names.
voice	Defines if the avatar is able or prefers to speak when used for OutputVerbalCommunication and understand when used for InputVerbalCommunication.
text	Defines if the avatar is able or prefers to write when used for OutputVerbalCommunication and read when used for InputVerbalCommunication.
language	Defines the preferred language for verbal communication. If it is not specified, the value of the attribute defaultLanguage defined in the CommunicationSkills type will be applied.

5.2.6.6 LanguageType

5.2.6.6.1 XML representation syntax

Diagram	
Source	<complexType name="LanguageType"> <attribute name="name" type="language" use="required"/> <attribute name="preference" type="vwoc:communicationPreferenceType" use="required"/> </complexType>

5.2.6.6.2 Binary representation syntax

LanguageType {	***Number of bits***	***Mnemonic***
name	See ISO 10646	UTF-8
preference		communicationPreferenceType
}		

5.2.6.6.3 Semantics

Name	*Definition*

LanguageType	Defines secondary communication skills for VerbalCommunication. In case it is not possible to use the preferred language (or the default language) defined for communicating with other avatar, these secondary languages will be applied.
name	String that specifies the name of the language (ex. en for English, es for Spanish…), according to the ISO 639 which describes the set of International Standards that lists short codes for language names.
preference	Define the preference for using the language in verbal communication: voice or text.

5.2.6.7 communicationPreferenceType

5.2.6.7.1 XML representation syntax

Source	<simpleType name="communicationPreferenceType"> <restriction base="string"> <enumeration value="voice"/> <enumeration value="text"/> </restriction> </simpleType>

5.2.6.7.2 Binary representation syntax

communicationPreferenceType {	***Number of bits***	***Mnemonic***
communicationPreference	1	bslbf
}		

5.2.6.7.3 Semantics

Name	*Definition*
communicationPreferenceType	Defines the preferred level of communication of the avatar: voice or text. The binary representation of the type is defined as follows. (0: voice, 1: text)

5.2.6.8 communicationPreferenceLevelType

5.2.6.8.1 XML representation syntax

Source	<simpleType name="communicationPreferenceLevelType"> <restriction base="string"> <enumeration value="preferred"/> <enumeration value="enabled"/> <enumeration value="disabled"/> </restriction> </simpleType>

5.2.6.8.2 Binary representation syntax

communicationPreferenceLevelType {	***Number of bits***	***Mnemonic***

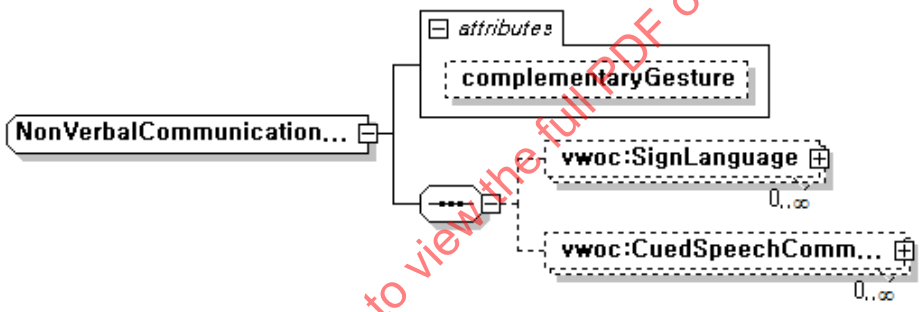
communicationPreferenceLevel	2	bslbf
}		

5.2.6.8.3 Semantics

Name	*Definition*
communicationPreferenceLevelType	Defines the level of preference for each language that the avatar can speak/understand. This level can be: preferred, enabled or disabled. The binary representation of the type is defined as follows. (0: preferred, 1: enabled, 2: disabled, or 3: reserved)

5.2.6.9 NonVerbalCommunicationType

5.2.6.9.1 XML representation syntax

Diagram	
Source	<complexType name="NonVerbalCommunicationType"> <sequence> <element name="SignLanguage" type="vwoc:SignLanguageType" minOccurs="0" maxOccurs="unbounded"/> <element name="CuedSpeechCommunication" type="vwoc:SignLanguageType" minOccurs="0" maxOccurs="unbounded"/> </sequence> <attribute name="complementaryGesture" type="string" use="optional"/> </complexType>

5.2.6.9.2 Binary representation syntax

NonVerbalCommunicationType {	***Number of bits***	***Mnemonic***
SignLanguageFlag	1	bslbf
CuedSpeechCommunicationFlag	1	bslbf
complementaryGestureFlag	1	bslbf
if(SignLanguageFlag) {		
NumSignLanguage		vluimsbf5

for(k=0; k<NumSignLanguage; k++){		
SignLanguage[k]		SignLanguageType
}		
}		
if(CuedSpeechCommunicationFlag) {		
NumCuedSpeechCommunication		vluimsbf5
for(k=0; k< NumCuedSpeechCommunication; k++){		
CuedSpeechCommunication[k]		SignLanguageType
}		
}		
if(complementaryGestureFlag) {		
complementaryGesture	See ISO 10646	UTF-8
}		
}		

5.2.6.9.3 Semantics

Name	*Definition*
NonVerbalCommunicationType	Specifies the avatar's non-verbal communication skills.
SignLanguageFlag	This field, which is only present in the binary representation, signals the presence of the SignLanguage elements. "1" means that the elements shall be used. "0" means that the elements shall not be used.
CuedSpeechCommunicationFlag	This field, which is only present in the binary representation, signals the presence of the CuedSpeechCommunication elements. "1" means that the elements shall be used. "0" means that the elements shall not be used.
complementaryGestureFlag	This field, which is only present in the binary representation, signals the presence of the complementaryGeature attribute. "1" means that the attribute shall be used. "0" means that the attribute shall not be used.
NumSingLanguage	This field, which is only present in the binary representation, specifies the number of SignLanguage elements contained in the SignLanguage.
SignLanguage	Defines the sign languages that the avatar is able to perform when used for OutputVerbalCommunication and interpret when used for InputVerbalCommunication.
NumCuedSpeechCommunication	This field, which is only present in the binary representation, specifies the number of CuedSpeechCommunication elements contained in the CuedSpeechCommunication.
CuedSpeechCommunication	Defines the cued speech communications that the avatar is able to perform when used for OutputVerbalCommunication and interpret when used for InputVerbalCommunication.
complementaryGesture	Defines if the avatar is able to perform complementary gesture during

	output verbal communication.

5.2.6.10 SignLanguageType

5.2.6.10.1 XML representation syntax

Diagram	
Source	<complexType name="SignLanguageType"> <attribute name="name" type="language" use="required"/> </complexType>

5.2.6.10.2 Binary representation syntax

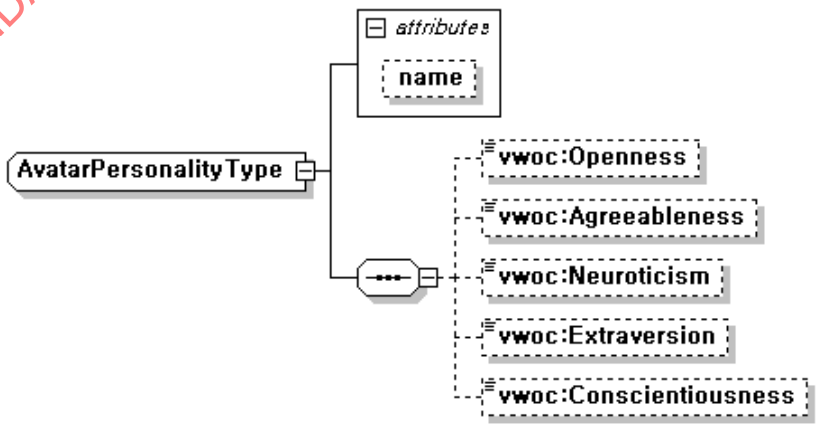
SignLanguageType {	*Number of bits*	*Mnemonic*
name	See ISO 10646	UTF-8
}		

5.2.6.10.3 Semantics

Name	*Definition*
`SignLanguageType`	Defines secondary communication skills for `NonVerbalCommunication` (sign or cued communication). In case it is not possible to use the preferred language (or the default language), these secondary languages will be applied.
`name`	Specifies the name of the language (ex. en for English, es for Spanish…), according to the ISO 639 which describes the set of International Standards that lists short codes for language names.

5.2.7 AvatarPersonalityType

5.2.7.1 XML representation syntax

Diagram	
Source	<complexType name="AvatarPersonalityType"> <sequence>

```
            <element name="Openness" type="mpeg7:minusOneToOneType" minOccurs="0"/>
            <element name="Agreeableness" type="mpeg7:minusOneToOneType" minOccurs="0"/>
            <element name="Neuroticism" type="mpeg7:minusOneToOneType" minOccurs="0"/>
            <element name="Extraversion" type="mpeg7:minusOneToOneType" minOccurs="0"/>
            <element name="Conscientiousness" type="mpeg7:minusOneToOneType" minOccurs="0"/>
        </sequence>
        <attribute name="name" type="string"/>
    </complexType>
```

5.2.7.2 Binary representation syntax

AvatarPersonalityType{	***Number of bits***	***Mnemonic***
OpennessFlag	1	bslbf
AgreeablenessFlag	1	bslbf
NeuroticismFlag	1	bslbf
ExtraversionFlag	1	bslbf
ConscientiousnessFlag	1	bslbf
NameFlag	1	bslbf
if(OpennessFlag){		
Openness	32	fsbf
}		
if(AgreeablenessFlag){		
Agreeableness	32	fsbf
}		
if(NeuroticismFlag){		
Neuroticism	32	fsbf
}		
if(ExtraversionFlag){		
Extraversion	32	fsbf
}		
if(ConscientiousnessFlag){		
Conscientiousness	32	fsbf
}		
if(NameFlag){		
name	See ISO 10646	UTF-8

}		
}		

5.2.7.3 Semantics

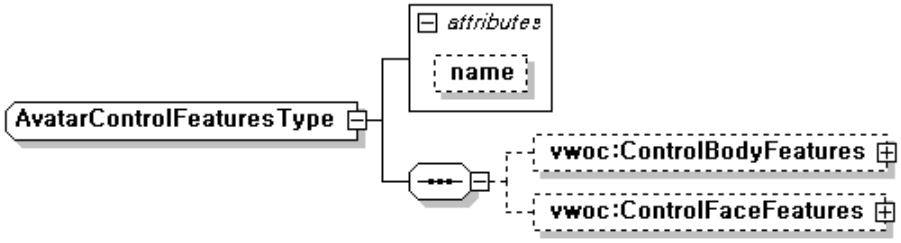
This tag (see Reference [3]) defines the personality of the avatar. This definition is based on the OCEAN model (see Referece [1]), consisting in a set of characteristics that personality is composed of. A combination of these characteristics is a specific personality. Therefore, an avatar contains a subtag for each attribute defined in OCEAN's model. They are: `openness`, `conscientiousness`, `extraversion`, `agreeableness` and `neuroticism`.

The purpose of this tag is to provide the possibility to define the avatar personality that is desired, and that the architecture of the virtual world can interpret as the inhabitant wishes. It would be able to adapt the avatar's verbal and non-verbal communication to this personality. Moreover, emotions and moods that could be provoked by virtual world events, avatar-avatar communication or the real time flow, will be modulated by this base personality.

Name	*Definition*
`AvatarPersonalityType`	A type that contains a set of descriptors defining the personality of the avatar.
`OpennessFlag`	This field, which is only present in the binary representation, signals the presence of the `Openness` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`AgreeablenessFlag`	This field, which is only present in the binary representation, signals the presence of the `Agreeableness` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`NeuroticismFlag`	This field, which is only present in the binary representation, signals the presence of the `Neuroticism` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`ExtraversionFlag`	This field, which is only present in the binary representation, signals the presence of the `Extraversion` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`ConscientiousnessFlag`	This field, which is only present in the binary representation, signals the presence of the `Conscientiousness` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`NameFlag`	This field, which is only present in the binary representation, signals the presence of the `name` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`Openness`	A value between -1 and 1 specifying the openness level of the personality.
`Agreeableness`	A value between -1 and 1 specifying the agreeableness level of the personality.
`Neuroticism`	A value between -1 and 1 specifying the neuroticism level of the personality.
`Extraversion`	A value between -1 and 1 specifying the extraversion level of the personality.
`Conscientiousness`	A value between -1 and 1 specifying the conscientiousness level of the personality.
`name`	A string value that specifies the name of personality.

5.2.8 AvatarControlFeaturesType

5.2.8.1 XML representation syntax

Diagram	
Source	<complexType name="AvatarControlFeaturesType"> <sequence> <element name="ControlBodyFeatures" type="vwoc:ControlBodyFeaturesType" minOccurs="0"/> <element name="ControlFaceFeatures" type="vwoc:ControlFaceFeaturesType" minOccurs="0"/> </sequence> <attribute name="name" type="string"/> </complexType>

5.2.8.2 Binary representation syntax

AvatarControlFeaturesType {	***Number of bits***	***Mnemonic***
ControlBodyFeaturesFlag	1	bslbf
ControlFaceFeaturesFlag	1	bslbf
if(ControlBodyFeaturesFlag){		
ControlBodyFeatures		ControlBodyFeaturesType
}		
if(ControlFaceFeaturesFlag){		
ControlFaceFeatures		ControlFaceFeaturesType
}		
if(NameFlag){		
name	See ISO 10646	UTF-8
}		
}		

5.2.8.3 Semantics

Name	*Description*
`AvatarControlFeaturesType`	A type that contains a set of descriptors defining possible place-holders for sensors on body skeleton and face feature points.
`ControlBodyFeaturesFlag`	This field, which is only present in the binary representation, signals the presence of the `ControlBodyFeatures` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.

ControlFaceFeaturesFlag	This field, which is only present in the binary representation, signals the presence of the ControlFaceFeatures element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
NameFlag	This field, which is only present in the binary representation, signals the presence of the name element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
ControlBodyFeatures	Set of elements that control moves of the body (bones).
ControlFaceFeatures	Set of elements that control moves of the face.
name	A string value that specifies the name of control features.

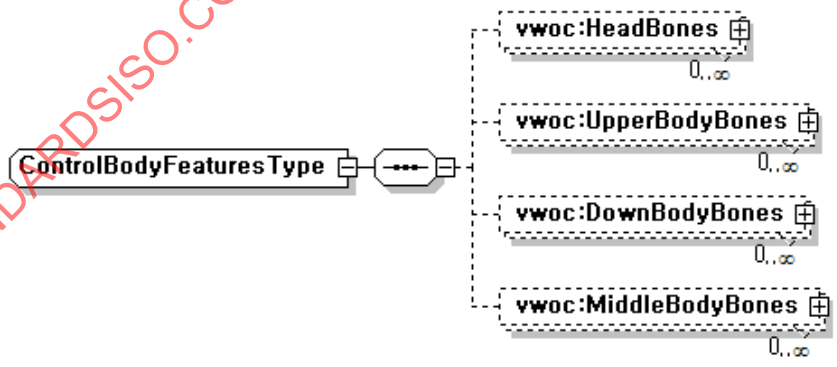
5.2.8.4 Examples

This example shows the description of controlling body and face features with the following semantics. The features control is given and works as a container.

```
<vwoc:ControlFeatures>
    <vwoc:ControlBodyFeatures>
        <vwoc:HeadBones name="urn:mpeg:mpeg-v:01-VWOC-HeadBonesCS-NS:skull"
alias="Head"/>
        …
    </vwoc:ControlBodyFeatures>
    <vwoc:ControlFaceFeatures>
        <vwoc:HeadOutline>
            …
        </vwoc:HeadOutline>
        …
    </vwoc:ControlFaceFeatures>
</vwoc:ControlFeatures>
```

5.2.8.5 ControlBodyFeaturesType

5.2.8.5.1 XML representation syntax

Diagram	
Source	<complexType name="ControlBodyFeaturesType"> <sequence> <element name="HeadBones" type="vwoc:ControlBodyFeaturesDescriptionType" minOccurs="0" maxOccurs="unbounded"/> <element name="UpperBodyBones" type="vwoc:ControlBodyFeaturesDescriptionType" minOccurs="0" maxOccurs="unbounded"/> <element name="DownBodyBones" type="vwoc:ControlBodyFeaturesDescriptionType" minOccurs="0" maxOccurs="unbounded"/> <element name="MiddleBodyBones" type="vwoc:ControlBodyFeaturesDescriptionType" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType>

5.2.8.5.2 Binary representation syntax

ControlBodyFeaturesType{	***Number of bits***	***Mnemonic***
HeadBonesFlag	1	bslbf
UpperBodyBonesFlag	1	bslbf
DownBodyBonesFlag	1	bslbf
MiddleBodyBonesFlag	1	bslbf
if(HeadBonesFlag){		
NumHeadBones		vluimsbf5
for(k=0; k<HeadBones; k++){		
HeadBones[k]		ControlBodyFeaturesDescriptionType
}		
}		
if(UpperBodyBonesFlag){		
NumUpperBodyBones		vluimsbf5
for(k=0; k<NumUpperBodyBones; k++){		
UpperBodyBones[k]		ControlBodyFeaturesDescriptionType
}		
}		
if(DownBodyBonesFlag){		
NumDownBodyBones		vluimsbf5
for(k=0; k<NumDownBodyBones; k++){		
DownBodyBones[k]		ControlBodyFeaturesDescriptionType
}		
}		
if(MiddleBodyBonesFlag){		
NumMiddleBodyBones		vluimsbf5

for(k=0; k<NumMiddleBodyBones; k++){		
MiddleBodyBones[k]		ControlBodyFeaturesDescriptionType
}		
}		
}		

5.2.8.5.3 Semantics

Name	*Description (Compare with Human Bones)*
ControlBodyFeaturesType	A type that contains a set of descriptors defining possible place-holders for sensors on body skeleton.
HeadBonesFlag	This field, which is only present in the binary representation, signals the presence of the HeadBones element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
UpperBodyBonesFlag	This field, which is only present in the binary representation, signals the presence of the UpperBodyBones element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
DownBodyBonesFlag	This field, which is only present in the binary representation, signals the presence of the DownBodyBones element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
MiddleBodyBonesFlag	This field, which is only present in the binary representation, signals the presence of the MiddleBodyBones element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
NumHeadBones	This field, which is only present in the binary representation, specifies the number of HeadBones elements contained in the ControlBodyFeaturesType.
HeadBones	Set of bones on the head: a list of the head bones is included in a classification scheme (CS) term. A CS that may be used for this purpose is the HeadBonesCS defined in ISO/IEC 23005-6, A.2.12.1. The binary representation of the HeadBonesCS is also defined in the same Annex.

name	*description*
cervicalVertebrae7	Cervical vertebrae 7
cervicalVertebrae6	Cervical vertebrae 6
cervicalVertebrae5	Cervical vertebrae 5
cervicalVertebrae4	Cervical vertebrae 4
cervicalVertebrae3	Cervical vertebrae 3
cervicalVertebrae2	Cervical vertebrae 2
cervicalVertebrae1	Cervical vertebrae 1
skull	Skull
lEyelid	Left eyelid
rEyelid	Right eyelid
lEyeball	Left eyeball
rEyeball	Right eyeball
lEyebrow	Left eyebrow
rEyebrow	Right eyebrow
jaw	Jaw

NumUpperBodyBones	This field, which is only present in the binary representation, specifies the number of UpperBodyBones elements contained in the ControlBodyFeaturesType.
UpperBodyBones	Set of bones on the upper part of the body, mainly arms and hands bones: a list of the upper body bones is included in a classification scheme (CS) term. A CS that may be used for this purpose is the UpperBodyBonesCS defined in

<table>
<tr><th>Name</th><th>Description (Compare with Human Bones)</th></tr>
<tr><td></td><td>ISO/IEC 23005-6, A.2.12.2. The binary representation of the UpperBodyBonesCS is also defined in the same Annex.
<table>
<tr><th>name</th><th>description</th></tr>
<tr><td>lClavicle</td><td>Left clavicle</td></tr>
<tr><td>lScapulae</td><td>Left scapulae</td></tr>
<tr><td>lHumerus</td><td>Left humerus</td></tr>
<tr><td>lRadius</td><td>Left radius</td></tr>
<tr><td>lWrist</td><td>Left wrist</td></tr>
<tr><td>lHand</td><td>Left hand</td></tr>
<tr><td>lThumb</td><td>Left thumb metacarpal</td></tr>
<tr><td>lPhalanges1</td><td>Left Phalanges1</td></tr>
<tr><td>lThumb2</td><td>Left thumb</td></tr>
<tr><td>lPhalanges2</td><td>Left Phalanges2</td></tr>
<tr><td>lIndex</td><td>Left index metacarpal</td></tr>
<tr><td>lPhalanges3</td><td>Left Phalanges3</td></tr>
<tr><td>lPhalanges4</td><td>Left Phalanges4</td></tr>
<tr><td>lPhalanges5</td><td>Left Phalanges5</td></tr>
<tr><td>lMiddle</td><td>Left middle metacarpal</td></tr>
<tr><td>lPhalanges6</td><td>Left Phalanges6</td></tr>
<tr><td>lPhalanges7</td><td>Left Phalanges7</td></tr>
<tr><td>lPhalanges8</td><td>Left Phalanges8</td></tr>
<tr><td>lRing</td><td>Left ring metacarpal</td></tr>
<tr><td>lPhalanges9</td><td>Left Phalanges9</td></tr>
<tr><td>lPhalanges10</td><td>Left Phalanges10</td></tr>
<tr><td>lPhalanges11</td><td>Left Phalanges11</td></tr>
<tr><td>lPinky</td><td>Left pinky metacarpal</td></tr>
<tr><td>lPhalanges12</td><td>Left Phalanges12</td></tr>
<tr><td>lPhalanges13</td><td>Left Phalanges13</td></tr>
<tr><td>lPhalanges14</td><td>Left Phalanges14</td></tr>
<tr><td>rClavicle</td><td>Right clavicle</td></tr>
<tr><td>rScapulae</td><td>Right scapulae</td></tr>
<tr><td>rHumerus</td><td>Right humerus</td></tr>
<tr><td>rRadius</td><td>Right radius</td></tr>
<tr><td>rWrist</td><td>Right wrist</td></tr>
<tr><td>rHand</td><td>Right hand</td></tr>
<tr><td>rThumb</td><td>Right thumb Metacarpal</td></tr>
<tr><td>rPhalanges1</td><td>Right Phalanges1</td></tr>
<tr><td>rThumb2</td><td>Right thumb</td></tr>
<tr><td>rPhalanges2</td><td>Right Phalanges2</td></tr>
<tr><td>rIndex</td><td>Right index metacarpal</td></tr>
<tr><td>rPhalanges3</td><td>Right Phalanges3</td></tr>
<tr><td>rPhalanges4</td><td>Right Phalanges4</td></tr>
<tr><td>rPhalanges5</td><td>Right Phalanges5</td></tr>
<tr><td>rMiddle</td><td>Right middle metacarpal</td></tr>
<tr><td>rPhalanges6</td><td>Right Phalanges6</td></tr>
<tr><td>rPhalanges7</td><td>Right Phalanges7</td></tr>
<tr><td>rPhalanges8</td><td>Right Phalanges8</td></tr>
<tr><td>rRing</td><td>Right ring metacarpal</td></tr>
<tr><td>rPhalanges9</td><td>Right Phalanges9</td></tr>
<tr><td>rPhalanges10</td><td>Right Phalanges10</td></tr>
<tr><td>rPhalanges11</td><td>Right Phalanges11</td></tr>
<tr><td>rPinky</td><td>Right pinky metacarpal</td></tr>
<tr><td>rPhalanges12</td><td>Right Phalanges12</td></tr>
<tr><td>rPhalanges13</td><td>Right Phalanges13</td></tr>
<tr><td>rPhalanges14</td><td>Right Phalanges14</td></tr>
</table></td></tr>
<tr><td>NumDownBodyBones</td><td>This field, which is only present in the binary representation, specifies the number of DownBodyBones elements contained in the ControlBodyFeaturesType.</td></tr>
<tr><td>DownBodyBones</td><td>Set of bones on the down part of the body, mainly legs and foot bones: a list of</td></tr>
</table>

<table>
<tr><th>Name</th><th>Description (Compare with Human Bones)</th></tr>
<tr><td></td><td>the down body bones is included in a classification scheme (CS) term. A CS that may be used for this purpose is the DownBodyBonesCS defined in ISO/IEC 23005-6, A.2.12.3. The binary representation of the DownBodyBonesCS is also defined in the same Annex.
<table>
<tr><th>name</th><th>Description</th></tr>
<tr><td>lFemur</td><td>Left femur</td></tr>
<tr><td>lPatella</td><td>Left patella (knee bone)</td></tr>
<tr><td>lTibia</td><td>Left tibia (femur in front)</td></tr>
<tr><td>lFibulae</td><td>Left fibulae</td></tr>
<tr><td>lTarsals1</td><td>Left tarsals1</td></tr>
<tr><td>lTarsals2</td><td>Left tarsals2 (7 are all)</td></tr>
<tr><td>lMetaTarsals</td><td>Left metatarsals (5) (foot parts)</td></tr>
<tr><td>lPhalanges</td><td>Left Phalanges (1 - 14) (foot parts)</td></tr>
<tr><td>rFemur</td><td>Right femur</td></tr>
<tr><td>rPatella</td><td>Right patella (knee bone)</td></tr>
<tr><td>rTibia</td><td>Right tibia (femur in front)</td></tr>
<tr><td>rFibulae</td><td>Right fibulae</td></tr>
<tr><td>rTarsals1</td><td>Right tarsals1 (parts of ankle)</td></tr>
<tr><td>rTarsals2</td><td>Right tarsals2 (7 are all)</td></tr>
<tr><td>rMetaTarsals</td><td>Right metatarsals (5) (foot parts)</td></tr>
<tr><td>rPhalanges</td><td>Right Phalanges (1 - 14) (foot parts)</td></tr>
</table></td></tr>
<tr><td>NumMiddleBodyBones</td><td>This field, which is only present in the binary representation, specifies the number of MiddleBodyBones elements contained in the ControlBodyFeaturesType.</td></tr>
<tr><td>MiddleBodyBones</td><td>Set of bones on the middle part of the body, torso: a list of the middle body bones is included in a classification scheme (CS) term. A CS that may be used for this purpose is the MiddleBodyBonesCS defined in ISO/IEC 23005-6, A.2.12.4. The binary representation of the MiddleBodyBonesCS is also defined in the same Annex.
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>sacrum</td><td>Sacrum</td></tr>
<tr><td>pelvis</td><td>Pelvis</td></tr>
<tr><td>lumbarVertebrae5</td><td>Lumbar vertebrae 5</td></tr>
<tr><td>lumbarVertebrae4</td><td>Lumbar vertebrae 4</td></tr>
<tr><td>lumbarVertebrae3</td><td>Lumbar vertebrae 3</td></tr>
<tr><td>lumbarVertebrae2</td><td>Lumbar vertebrae 2</td></tr>
<tr><td>lumbarVertebrae1</td><td>Lumbar vertebrae 1</td></tr>
<tr><td>thoracicVertebrae12</td><td>Thoracic vertebrae 12</td></tr>
<tr><td>thoracicVertebrae11</td><td>Thoracic vertebrae 11</td></tr>
<tr><td>thoracicVertebrae10</td><td>Thoracic vertebrae 10</td></tr>
<tr><td>thoracicVertebrae9</td><td>Thoracic vertebrae 9</td></tr>
<tr><td>thoracicVertebrae8</td><td>Thoracic vertebrae 8</td></tr>
<tr><td>thoracicVertebrae7</td><td>Thoracic vertebrae 7</td></tr>
<tr><td>thoracicVertebrae6</td><td>Thoracic vertebrae 6</td></tr>
<tr><td>thoracicVertebrae5</td><td>Thoracic vertebrae 5</td></tr>
<tr><td>thoracicVertebrae4</td><td>Thoracic vertebrae 4</td></tr>
<tr><td>thoracicVertebrae3</td><td>Thoracic vertebrae 3</td></tr>
<tr><td>thoracicVertebrae2</td><td>Thoracic vertebrae 2</td></tr>
<tr><td>thoracicVertebrae1</td><td>Thoracic vertebrae 1</td></tr>
</table></td></tr>
</table>

5.2.8.5.4 Examples

This example shows the description of controlling body features with the following semantics. The body features control maps the user defined body feature points to the placeholders. The following set of the feature points are mapped to the placeholders defined in the semantics.

Name of Placeholder	User defined features
sacrum	Hip
pelvis	Abdomen
lFemur	LThigh
lTibia (femur in front)	LShin
lFibulae	LFoot
rFemur	RThigh
rTibia (femur in front)	RShin
rFibulae	RFoot
thoracicVertebrae1	Chest
cervicalVertebrae1	Neck
skull	Head
lClavicle	LCollar
lHumerus	LShldr
lRadius	LForeArm
lHand	LHand
rClavicle	RCollar
rHumerus	RShldr
rRadius	RForeArm
rHand	RHand

```
<vwoc:ControlFeatures>
   <vwoc:ControlBodyFeatures>
     <vwoc:HeadBones name="urn:mpeg:mpeg-v:01-VWOC-HeadBonesCS-NS:skull"
alias="Head"/>
     <vwoc:HeadBones name="urn:mpeg:mpeg-v:01-VWOC-HeadBonesCS-
NS:cervicalVerbae1" alias="Neck"/>
     <vwoc:UpperBodyBones name="urn:mpeg:mpeg-v:01-VWOC-UpperBodyBonesCS-
NS:lClavicle" alias="LCollar"/>
     <vwoc:UpperBodyBones name="urn:mpeg:mpeg-v:01-VWOC-UpperBodyBonesCS-
NS:lHumerus" alias="LShldr"/>
     <vwoc:UpperBodyBones name="urn:mpeg:mpeg-v:01-VWOC-UpperBodyBonesCS-
NS:lRadius" alias="LForeArm"/>
     <vwoc:UpperBodyBones name="urn:mpeg:mpeg-v:01-VWOC-UpperBodyBonesCS-
NS:lHand" alias="LHand"/>
     <vwoc:UpperBodyBones name="urn:mpeg:mpeg-v:01-VWOC-UpperBodyBonesCS-
NS:rClavicle" alias="RCollar"/>
     <vwoc:UpperBodyBones name="urn:mpeg:mpeg-v:01-VWOC-UpperBodyBonesCS-
NS:rHumerus" alias="RShldr"/>
     <vwoc:UpperBodyBones name="urn:mpeg:mpeg-v:01-VWOC-UpperBodyBonesCS-
NS:rRadius" alias="RForeArm"/>
     <vwoc:UpperBodyBones name="urn:mpeg:mpeg-v:01-VWOC-UpperBodyBonesCS-
NS:rHand" alias="RHand"/>
     <vwoc:DownBodyBones name="urn:mpeg:mpeg-v:01-VWOC-DownBodyBonesCS-
NS:lFemur" alias="LThigh"/>
     <vwoc:DownBodyBones name="urn:mpeg:mpeg-v:01-VWOC-DownBodyBonesCS-
NS:lTibia" alias="LShin"/>
     <vwoc:DownBodyBones name="urn:mpeg:mpeg-v:01-VWOC-DownBodyBonesCS-
NS:lFibulae" alias="LFoot"/>
     <vwoc:DownBodyBones name="urn:mpeg:mpeg-v:01-VWOC-DownBodyBonesCS-
NS:rFemur" alias="RThigh"/>
```



```
      <vwoc:DownBodyBones name="urn:mpeg:mpeg-v:01-VWOC-DownBodyBonesCS-
NS:rTibia" alias="RShin"/>
      <vwoc:DownBodyBones name="urn:mpeg:mpeg-v:01-VWOC-DownBodyBonesCS-
NS:rFibulae" alias="RFoot"/>
      <vwoc:MiddleBodyBones name="urn:mpeg:mpeg-v:01-VWOC-MiddleBodyBonesCS-
NS:sacrum" alias="Hip"/>
      <vwoc:MiddleBodyBones name="urn:mpeg:mpeg-v:01-VWOC-MiddleBodyBonesCS-
NS:pelvis" alias="Abdomen"/>
      <vwoc:MiddleBodyBones name="urn:mpeg:mpeg-v:01-VWOC-MiddleBodyBonesCS-
NS:thoracicVertebrae1" alias="Chest"/>
   </vwoc:ControlBodyFeatures>
</vwoc:ControlFeatures>
```

5.2.8.6 ControlBodyFeaturesDescriptionType

5.2.8.6.1 XML representation syntax

Diagram	

Source	`<complexType name="ControlBodyFeaturesDescriptionType">`
	`  <attribute name="name" type="mpeg7:termReferenceType" use="required"/>`
	`  <attribute name="alias" type="string" use="required"/>`
	`</complexType>`

5.2.8.6.2 Binary representation syntax

ControlBodyFeaturesDescriptionType {	***Number of bits***	***Mnemonic***
TypeOfBodyFeature	3	bslbf
name	8	Number of bits are defined by the type of body feature as a reference to classification scheme
alias	See ISO 10646	UTF-8
}		

5.2.8.6.3 Semantics

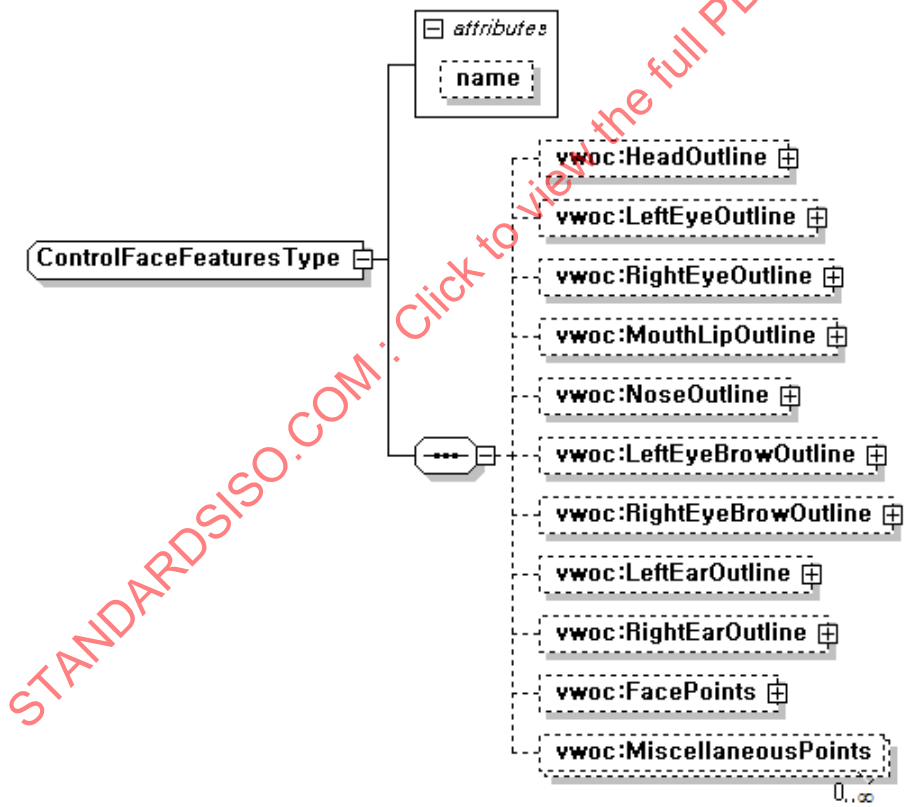
Name	*Definition*
`ControlBodyFeaturesDescriptionType`	A type that contains the name and its alias of a body feature.
`TypeOfBodyFeature`	This field, which is only present in the binary representation, describes a type of body features as one of the classification schemes (CSs). The CSs that may be used for this purpose is defined in ISO/IEC 23005-6, A.2.12.

	Type of Body Feature	*Binary representation for sensor type (3 bits)*
	HeadBonesCS	000
	UpperBodyBonesCS	001
	DownBodyBonesCS	010
	MiddleBodyBonesCS	011
	Reserved	100-111
name	Describes a type of body features as a reference to classification scheme (CS) term. The CSs that may be used for this purpose is defined in ISO/IEC 23005-6, A.2.12.	
alias	Describes the name of a specific body feature type.	

5.2.8.7 ControlFaceFeaturesType

5.2.8.7.1 XML representation syntax

Diagram



Source

```
<complexType name="ControlFaceFeaturesType">
    <sequence>
        <element name="HeadOutline" type="vwoc:OutlineType" minOccurs="0"/>
        <element name="LeftEyeOutline" type="vwoc:OutlineType" minOccurs="0"/>
        <element name="RightEyeOutline" type="vwoc:OutlineType" minOccurs="0"/>
        <element name="MouthLipOutline" type="vwoc:OutlineType" minOccurs="0"/>
        <element name="NoseOutline" type="vwoc:OutlineType" minOccurs="0"/>
        <element name="LeftEyeBrowOutline" type="vwoc:Outline4PointsType" minOccurs="0"/>
        <element name="RightEyeBrowOutline" type="vwoc:Outline4PointsType" minOccurs="0"/>
        <element name="LeftEarOutline" type="vwoc:Outline4PointsType" minOccurs="0"/>
        <element name="RightEarOutline" type="vwoc:Outline4PointsType" minOccurs="0"/>
        <element name="FacePoints" type="vwoc:OutlineType" minOccurs="0"/>
```

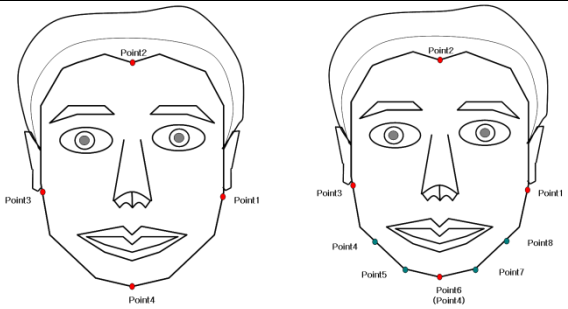
```
            <element name="MiscellaneousPoints" type="vwoc:PointType" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <attribute name="name" type="string"/>
</complexType>
```

5.2.8.7.2 Binary representation syntax

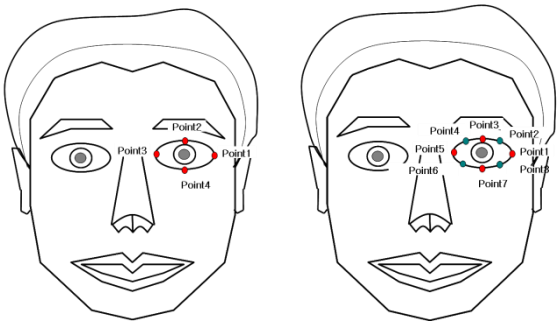
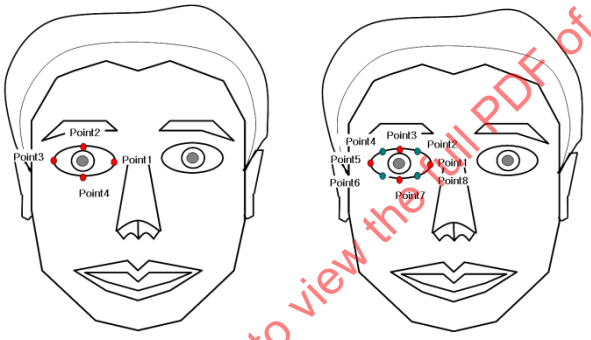
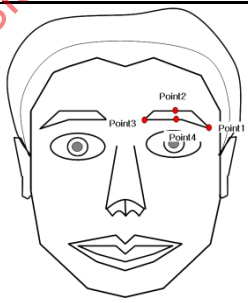
ControlFaceFeaturesType {	*Number of bits*	*Mnemonic*
HeadOutlineFlag	1	bslbf
LeftEyeOutlineFlag	1	bslbf
RightEyeOutlineFlag	1	bslbf
MouthLipOutlineFlag	1	bslbf
NoseOutlineFlag	1	bslbf
LeftEyeBrowOutlineFlag	1	bslbf
RightEyeBrowOutlineFlag	1	bslbf
LeftEarOutlineFlag	1	bslbf
RightEarOutlineFlag	1	bslbf
FacePointsFlag	1	bslbf
MiscellaneousPointsFlag	1	bslbf
NameFlag	1	bslbf
if(HeadOutlineFlag){		
HeadOutline		OutlineType
}		
if(LeftEyeOutlineFlag){		
LeftEyeOutline		OutlineType
}		
if(RightEyeOutlineFlag){		
RightEyeOutline		OutlineType
}		
if(MouthLipOutlineFlag){		
MouthLipOutline		OutlineType

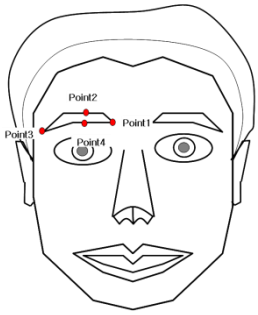
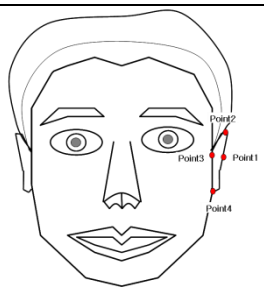
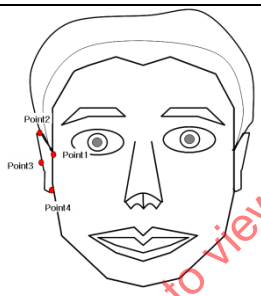
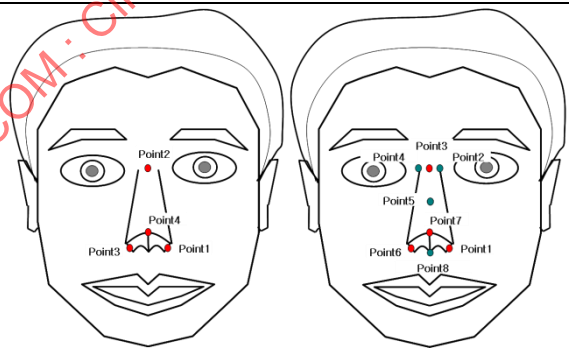
}		
if(NoseOutlineFlag){		
NoseOutline		OutlineType
}		
if(LeftEyeBrowOutlineFlag){		
LeftEyeBrowOutline		Outline4PointsType
}		
if(RightEyeBrowOutlineFlag){		
RightEyeBrowOutline		Outline4PointsType
}		
if(LeftEarOutlineFlag){		
LeftEarOutline		Outline4PointsType
}		
if(RightEarOutlineFlag){		
RightEarOutline		Outline4PointsType
}		
if(FacePointsFlag){		
FacePoints		OutlineType
}		
if(MiscellaneousPointsFlag){		
LoopMiscellaneousPoints		vluimsbf5
for(k=0;k< LoopMiscellaneousPoints;k++){		
MiscellaneousPoints[k]		PointType
}		
}		
if(NameFlag){		
name	See ISO 10646	UTF-8
}		
}		

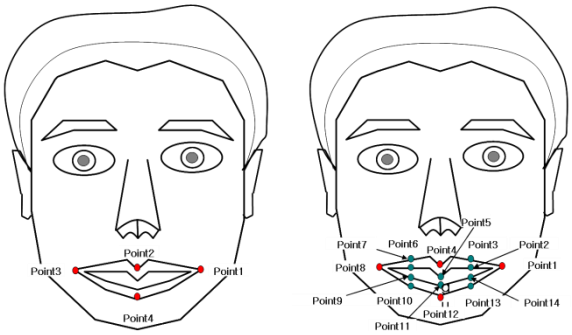
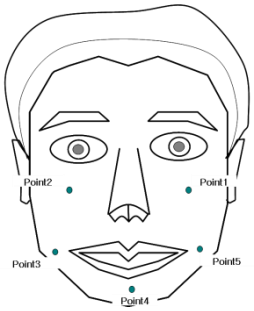
5.2.8.7.3 Semantics

Name	*Description*
`ControlFaceFeaturesType`	A type that contains the name and its alias of a face feature.
`HeadOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `HeadOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`LeftEyeOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `LeftEyeOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`RightEyeOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `RightEyeOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`MouthLipOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `MouthLipOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`NoseOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `NoseOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`LeftEyeBrowOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `LeftEyeBrowOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`RightEyeBrowOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `RightEyeBrowOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`LeftEarOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `LeftEarOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`RightEarOutlineFlag`	This field, which is only present in the binary representation, signals the presence of the `RightEarOutline` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`FacePointsFlag`	This field, which is only present in the binary representation, signals the presence of the `FacePoints` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`MiscellaneousPointsFlag`	This field, which is only present in the binary representation, signals the presence of the `MiscellaneousPoints` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`NameFlag`	This field, which is only present in the binary representation, signals the presence of the `Name` attribute. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`HeadOutline`	 Describes the outline of the head. The red dots in the figure on the left-hand side represent the points forming the basic outline. The additional 4 green points and the red dots on the right-hand side in the above figure form the high-resolution outline of the head.

Name	*Description*
`Outline4points`	Describes a basic outline of the head.

	`Outline8points`	Describes the extended outline of the head for the higher resolution outline of the head with 8 points.
`LeftEyeOutline`	 Describes the outline of the left eye. The red dots in the figure on the left-hand side represent the points forming the basic outline. The additional 4 green points and the red dots in the above figure on the right-hand side form the high-resolution outline. *Name* — *Description* `Outline4points` — Describes a basic outline of the left eye. `Outline8points` — Describes the extended outline of the left for the higher resolution outline of the head with 8 points.	
`RightEyeOutline`	 Describes the outline of the right eye. The red dots in the figure on the left-hand side represent the points forming the basic outline. The additional 4 green points and the red dots in the above figure on the right-hand side form the high-resolution outline. *Name* — *Description* `Outline4points` — Describes a basic outline of the right eye. `Outline8points` — Describes the extended outline of the left for the higher resolution outline of the head with 8 points.	
`LeftEyeBrowOutline`	 Describes the outline of the left eyebrow.	

RightEyeBrowOutline	 Describes the outline of the right eyebrow.
LeftEarOutline	 Describes the outline of the left ear.
RightEarOutline	 Describes the outline of the right ear.
NoseOutline	 Describes the basic outline of the nose. The red dots represent the points forming the basic outline. The red dots in the figure on the left-hand side represent the points forming the basic outline. The additional 4 green points and the red dots in the above figure on the right-hand side form the high-resolution outline. *Name* — *Description* `Outline4points` — Describes a basic outline of the nose. `Outline8points` — Describes the extended outline of the left for the higher resolution outline of the nose with 8 points.

MouthLipOutline	 Describes the outline of the mouth lips. The red dots represent the points forming the basic outline. The red dots in the figure on the left-hand side represent the points forming the basic outline. The additional 10 green points and the red dots in the above figure on the right-hand side form the high-resolution outline. *Name* — *Description* `Outline4points` — Describes a basic outline of the mouth lips. `Outline14points` — Describes the extended outline of the left for the higher resolution outline of the head with 14 points.
FacePoints	 The green dots form a high-resolution facial expression.
`LoopMiscellaneousPoints`	This field, which is only present in the binary representation, specifies the number of miscellaneous points.
`MiscellaneousPoints`	Describes any arbitrary feature points which can be placed and defined for an advanced facial feature control.
`name`	The name of the face control configuration.
`PointType`	An astract type providing root for two different point types, which are `LogicalPointType` and `Physical3DPointType` for specifying a feature point for face feature control.

5.2.8.7.4 Examples

This example shows the description of controlling face features with the following semantics. The face features control maps the user defined face feature points to the placeholders. The following sets of the feature points are mapped to the placeholders defined in the semantics.

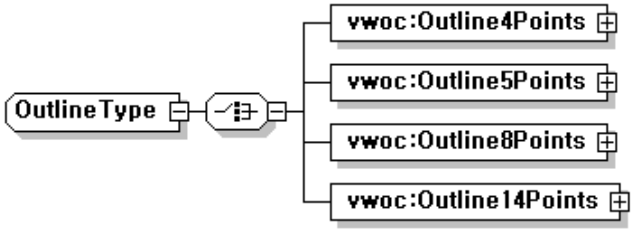
Name of Placeholder		User defined features	
HeadOutline	Point1	Head	HeadLeft
	Point2		HeadTop
	Point3		HeadRight
	Point4		HeadDown
LeftEyeOutline	Point1	Leye	LeyeLeft
	Point2		LeyeTop
	Point3		LeyeRight
	Point4		LeyeDown
RightEyeOutline	Point1	Reye	ReyeLeft
	Point2		ReyeTop
	Point3		ReyeRight
	Point4		ReyeDown

MouthLipOutline	Point1	Lips	LipsLeft
	Point2		LipsTop
	Point3		LipsRight
	Point4		LipsDown
NoseOutline	Point1	Nose	NoseLeft
	Point2		NoseTop
	Point3		NoseRight
	Point4		NoseDown

```
<vwoc:ControlFaceFeatures name="LogicalPointBasedFace">
   <vwoc:HeadOutline>
      <vwoc:Outline4Points>
         <vwoc:Point1 xsi:type="vwoc:LogicalPointType" name="HeadLeft"/>
         <vwoc:Point2 xsi:type="vwoc:LogicalPointType" name="HeadTop"/>
         <vwoc:Point3 xsi:type="vwoc:LogicalPointType" name="HeadRight"/>
         <vwoc:Point4 xsi:type="vwoc:LogicalPointType" name="HeadDown"/>
      </vwoc:Outline4Points>
   </vwoc:HeadOutline>
   <vwoc:LeftEyeOutline>
      <vwoc:Outline4Points>
         <vwoc:Point1 xsi:type="vwoc:LogicalPointType" name="LeyeLeft"/>
         <vwoc:Point2 xsi:type="vwoc:LogicalPointType" name="LeyeTop"/>
         <vwoc:Point3 xsi:type="vwoc:LogicalPointType" name="LeyeRight"/>
         <vwoc:Point4 xsi:type="vwoc:LogicalPointType" name="LeyeDown"/>
      </vwoc:Outline4Points>
   </vwoc:LeftEyeOutline>
   <vwoc:RightEyeOutline>
      <vwoc:Outline4Points>
         <vwoc:Point1 xsi:type="vwoc:LogicalPointType" name="ReyeLeft"/>
         <vwoc:Point2 xsi:type="vwoc:LogicalPointType" name="ReyeTop"/>
         <vwoc:Point3 xsi:type="vwoc:LogicalPointType" name="ReyeRight"/>
         <vwoc:Point4 xsi:type="vwoc:LogicalPointType" name="ReyeDown"/>
      </vwoc:Outline4Points>
   </vwoc:RightEyeOutline>
   <vwoc:MouthLipOutline>
      <vwoc:Outline4Points>
         <vwoc:Point1 xsi:type="vwoc:LogicalPointType" name="LipsLeft"/>
         <vwoc:Point2 xsi:type="vwoc:LogicalPointType" name="LipsTop"/>
         <vwoc:Point3 xsi:type="vwoc:LogicalPointType" name="LipsRight"/>
         <vwoc:Point4 xsi:type="vwoc:LogicalPointType" name="LipsDown"/>
      </vwoc:Outline4Points>
   </vwoc:MouthLipOutline>
   <vwoc:NoseOutline>
      <vwoc:Outline4Points>
         <vwoc:Point1 xsi:type="vwoc:LogicalPointType" name="NoseLeft"/>
         <vwoc:Point2 xsi:type="vwoc:LogicalPointType" name="NoseTop"/>
         <vwoc:Point3 xsi:type="vwoc:LogicalPointType" name="NoseRight"/>
         <vwoc:Point4 xsi:type="vwoc:LogicalPointType" name="NoseDown"/>
      </vwoc:Outline4Points>
   </vwoc:NoseOutline>
</vwoc:ControlFaceFeatures>
```

5.2.8.7.5 OutlineType

5.2.8.7.5.1 XML representation syntax

Diagram	
Source	<complexType name="OutlineType"> <choice> <element name="Outline4Points" type="vwoc:Outline4PointsType"/> <element name="Outline5Points" type="vwoc:Outline5PointsType"/> <element name="Outline8Points" type="vwoc:Outline8PointsType"/> <element name="Outline14Points" type="vwoc:Outline14PointsType"/> </choice> </complexType>

5.2.8.7.5.2 Binary representation syntax

OutlineType {	*Number of bits*	*Mnemonic*
OutlineTypeSelect	3	bslbf
if(OutlineTypeSelect ==0){		
Outline4Points		Outline4PointsType
}else if(OutlineTypeSelect ==1){		
Outline5Points		Outline5PointsType
}else if(OutlineTypeSelect ==2){		
Outline8Points		Outline8PointsType
}else if(OutlineTypeSelect ==3){		
Outline14Points		Outline14PointsType
}		
}		

5.2.8.7.5.3 Semantics

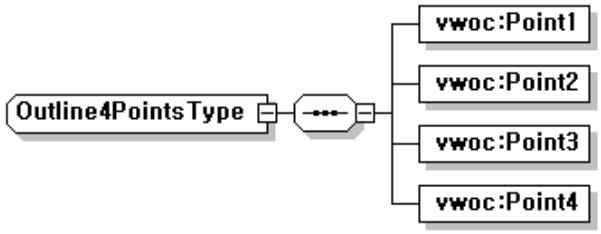
The OutlineType contains 4 different types of outline dependent upon the number of points forming the outline.

Name	*Description*
`OutlineType`	A type that describes the outline of each facial feature.
`OutlineTypeSelect`	This field, which is only present in the binary representation, determines the outline type with the number of points.

	(0: Outline4Points, 1:Outline5Points, 2: Outline8Points, 3: Outline14Points, 4-7: reserved)
Outline4Points	The outline with 4 points
Outline5Points	The outline with 5 points
Outline8Points	The outline with 8 points
Outline14Points	The outline with 14 points

5.2.8.7.5.4 Outline4PointsType

5.2.8.7.5.4.1 XML representation syntax

Diagram	
Source	<complexType name="Outline4PointsType"> <sequence> <element name="Point1" type="vwoc:PointType"/> <element name="Point2" type="vwoc:PointType"/> <element name="Point3" type="vwoc:PointType"/> <element name="Point4" type="vwoc:PointType"/> </sequence> </complexType>

5.2.8.7.5.4.2 Binary representation syntax

Outline4PointsType{	*Number of bits*	*Mnemonic*
Point1		PointType
Point2		PointType
Point3		PointType
Point4		PointType
}		

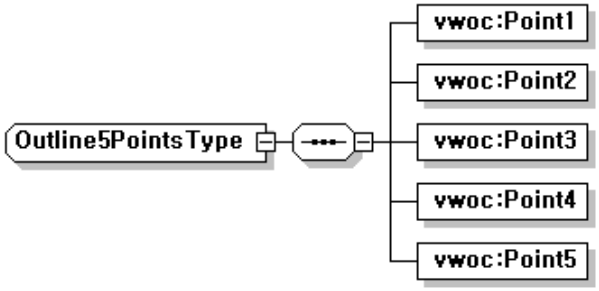
5.2.8.7.5.4.3 Semantics

The points are numbered counter-clockwise from the leftmost point. For example, if there are 4 points at the left, top, right, bottom of the outline, they are Point1, Point2, Point3, Point4, respectively.

Name	*Description*
Outline4PointsType	A type that describes the outline of each facial feature with four points.
Point1	The 1st point of the outline
Point2	The 2nd point of the outline
Point3	The 3rd point of the outline
Point4	The 4th point of the outline

5.2.8.7.5.5 Outline5PointsType

5.2.8.7.5.5.1 XML representation syntax

Diagram	
Source	<complexType name="Outline5PointsType"> <sequence> <element name="Point1" type="vwoc:PointType"/> <element name="Point2" type="vwoc:PointType"/> <element name="Point3" type="vwoc:PointType"/> <element name="Point4" type="vwoc:PointType"/> <element name="Point5" type="vwoc:PointType"/> </sequence> </complexType>

5.2.8.7.5.5.2 Binary representation syntax

Outline5PointsType{	*Number of bits*	*Mnemonic*
Point1		PointType
Point2		PointType
Point3		PointType
Point4		PointType
Point5		PointType
}		

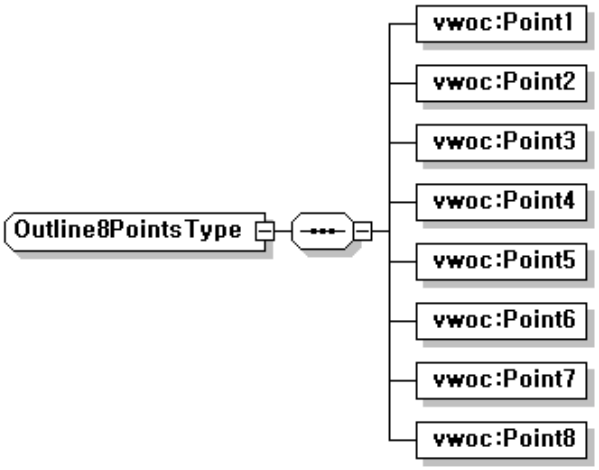
5.2.8.7.5.5.3 Semantics

The points are numbered counter-clockwise from the leftmost point. For the details, refer to the figure of FacePoints in 5.2.8.7.2.

Name	*Description*
`Outline5PointsType`	A type that describes the outline of each facial feature with five points.
`Point1`	The 1st point of the outline
`Point2`	The 2nd point of the outline
`Point3`	The 3rd point of the outline
`Point4`	The 4th point of the outline
`Point5`	The 5th point of the outline

5.2.8.7.5.6 Outline8PointsType

5.2.8.7.5.6.1 XML representation syntax

Diagram	

Source	`<complexType name="Outline8PointsType">`
	`<sequence>`
	`<element name="Point1" type="vwoc:PointType"/>`
	`<element name="Point2" type="vwoc:PointType"/>`
	`<element name="Point3" type="vwoc:PointType"/>`
	`<element name="Point4" type="vwoc:PointType"/>`
	`<element name="Point5" type="vwoc:PointType"/>`
	`<element name="Point6" type="vwoc:PointType"/>`
	`<element name="Point7" type="vwoc:PointType"/>`
	`<element name="Point8" type="vwoc:PointType"/>`
	`</sequence>`
	`</complexType>`

5.2.8.7.5.6.2 Binary representation syntax

Outline8PointsType{	*Number of bits*	*Mnemonic*
Point1		PointType
Point2		PointType
Point3		PointType
Point4		PointType
Point5		PointType
Point6		PointType
Point7		PointType
Point8		PointType
}		

5.2.8.7.5.6.3 Semantics

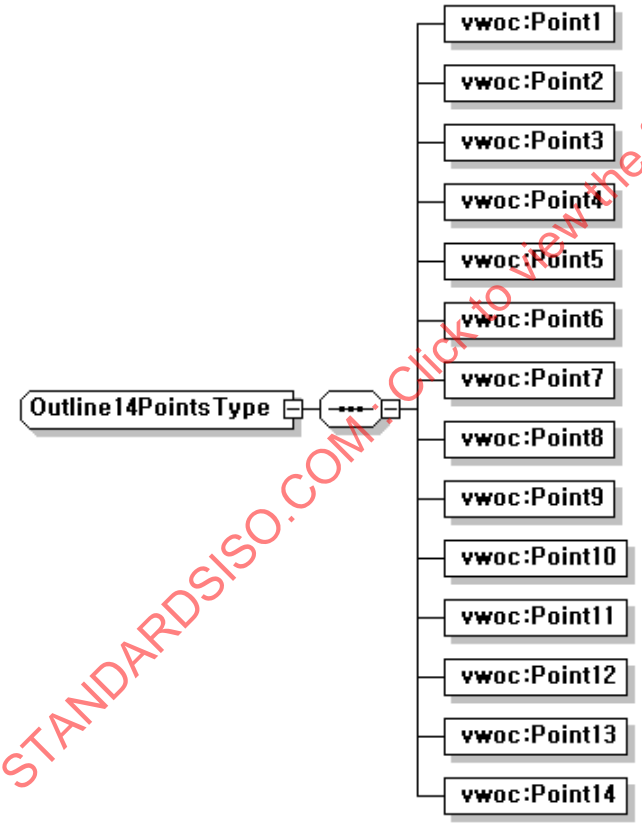
The points are numbered counter-clockwise from the leftmost point. For the details, refer to the figure of LeftEye in 5.2.8.7.2.

Name	*Description*
`Outline8PointsType`	A type that describes the outline of each facial feature with 8 points.
`Point1`	The 1st point of the outline
`Point2`	The 2nd point of the outline
`Point3`	The 3rd point of the outline
`Point4`	The 4th point of the outline
`Point5`	The 5th point of the outline
`Point6`	The 6th point of the outline
`Point7`	The 7th point of the outline
`Point8`	The 8th point of the outline

5.2.8.7.5.7 Outline14PointsType

5.2.8.7.5.7.1 XML representation syntax

Diagram



Source

```
<complexType name="Outline14PointsType">
    <sequence>
        <element name="Point1" type="vwoc:PointType"/>
        <element name="Point2" type="vwoc:PointType"/>
        <element name="Point3" type="vwoc:PointType"/>
        <element name="Point4" type="vwoc:PointType"/>
        <element name="Point5" type="vwoc:PointType"/>
        <element name="Point6" type="vwoc:PointType"/>
        <element name="Point7" type="vwoc:PointType"/>
        <element name="Point8" type="vwoc:PointType"/>
        <element name="Point9" type="vwoc:PointType"/>
        <element name="Point10" type="vwoc:PointType"/>
        <element name="Point11" type="vwoc:PointType"/>
        <element name="Point12" type="vwoc:PointType"/>
```



```
            <element name="Point13" type="vwoc:PointType"/>
            <element name="Point14" type="vwoc:PointType"/>
        </sequence>
    </complexType>
```

5.2.8.7.5.7.2 Binary representation syntax

Outline14PointsType{	*Number of bits*	*Mnemonic*
Point1		PointType
Point2		PointType
Point3		PointType
Point4		PointType
Point5		PointType
Point6		PointType
Point7		PointType
Point8		PointType
Point9		PointType
Point10		PointType
Point11		PointType
Point12		PointType
Point13		PointType
Point14		PointType
}		

5.2.8.7.5.7.3 Semantics

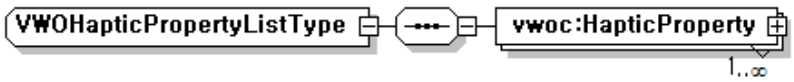
The points are numbered counter-clockwise from the leftmost point. For the details, refer to the figure of MouthLips in 5.2.8.7.2.

Name	*Description*
`Outline14PointsType`	A type that describes the outline of each facial feature with fourteen points.
`Point1`	The 1st point of the outline
`Point2`	The 2nd point of the outline
`Point3`	The 3rd point of the outline
`Point4`	The 4th point of the outline
`Point5`	The 5th point of the outline
`Point6`	The 6th point of the outline
`Point7`	The 7th point of the outline
`Point8`	The 8th point of the outline
`Point9`	The 9th point of the outline
`Point10`	The 10th point of the outline

Point11	The 11th point of the outline
Point12	The 12th point of the outline
Point13	The 13th point of the outline
Point14	The 14th point of the outline

5.2.9 VWOHapticPropertyListType

5.2.9.1 XML representation syntax

Diagram	
Source	<complexType name="VWOHapticPropertyListType"> <sequence> <element name="HapticProperty" type="vwoc:VWOHapticPropertyType" maxOccurs="unbounded"/> </sequence> </complexType>

5.2.9.2 Binary representation syntax

VWOHapticPropertyListType {	***Number of bits***	***Mnemonic***
NumVWOHapticPropertyType		vluimsbf5
for(k=0; k< NumVWOHapticPropertyType; k++){		
HapticProperty[k]		VWOHapticPropertyType
}		
}		

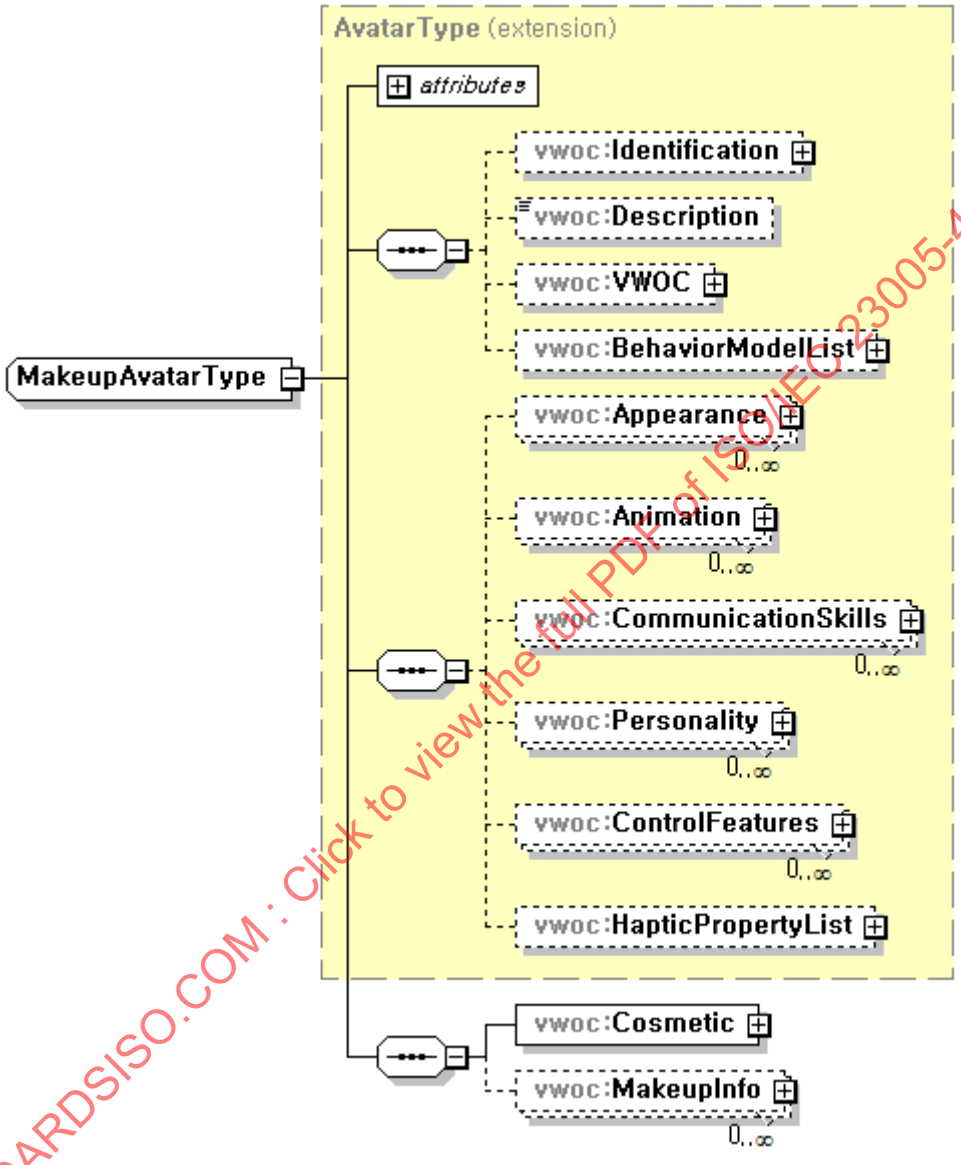
5.2.9.3 Semantics

Name	*Definition*
VWOHapticPropertyListType	Wrapper element type which allows multiple occurrences of the haptic properties associated to the virtual world object.
NumVWOHapticPropertyType	This field, which is only present in the binary representation, specifies the number of haptic property information contained in the haptic property list type.
HapticProperty	This element contains a set of high level descriptors of the haptic properties defined in the VWOHapticPropertyType of the virtual world object.

5.3 MakeupAvatarType

5.3.1 XML representation syntax

Diagram



Source

```
<complexType name="MakeupAvatarType">
    <complexContent>
        <extension base="vwoc:AvatarType">
            <sequence>
                <element name="Cosmetic" type="vwoc:CosmeticType"/>
                <element name="MakeupInfo" type="vwoc:MakeupInfoType" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```


5.3.2 Binary representation syntax

MakeupAvatarType {	***Number of bits***	***Mnemonic***

Avatar		AvatarType
MakeupInfoFlag	1	bslbf
Cosmetic		CosmeticType
if(MakeupInfoFlag) {		
numOfMakeupInfo		vluimsbf5
for(k=0;k<numOfMakeupInfo;k++) {		
MakeupInfo[k]		MakeupInfoType
}		
}		
}		

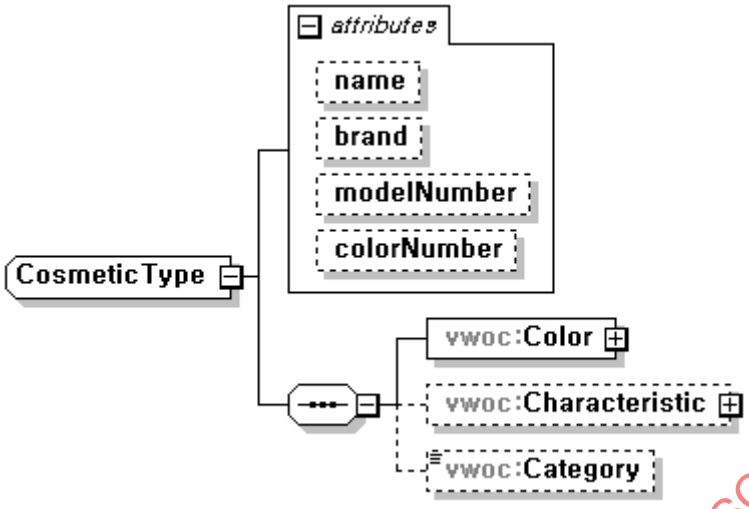
5.3.3 Semantics

Name	*Definition*
`MakeupAvatarType`	Tool for describing a makeup avatar. This type is extended from the Avatar Type.
`Cosmetic`	Describes cosmetic information.
`MakeupInfo`	Describes the makeup information of the makeup avatar.
`Avatar`	Contains the base type defined by `AvatarType`.
`MakeupInfoFlag`	This field, which is only present in the binary representation, signals the presence of the `MakeupInfo` elements. "1" means that the elements shall be used. "0" means that the elements shall not be used.
`numOfMakeupInfo`	This field, which is only present in the binary representation, specifies the number of `MakeupInfo` information contained in the makeup information for avatar.

5.3.4 CosmeticType

5.3.4.1 XML representation syntax

Diagram



Source

```
<complexType name="CosmeticType">
      <sequence>
            <element name="Color" type="vwoc:CosmeticSpectrumType"/>
            <element name="Characteristic" type="vwoc:CosmeticCharacteristicType" minOccurs="0"/>
            <element name="Category" type="mpeg7:termReferenceType" minOccurs="0"/>
      </sequence>
      <attribute name="name" type="string" use="optional"/>
      <attribute name="brand" type="string" use="optional"/>
      <attribute name="modelNumber" type="string" use="optional"/>
      <attribute name="colorNumber" type="string" use="optional"/>
</complexType>

<complexType name="CosmeticSpectrumType">
      <sequence>
            <choice>
                  <element name="Spectra" type="mpeg7:DoubleMatrixType"/>
                  <element name="SpectraURI" type="anyURI"/>
            </choice>
            <element name="CosmeticSpectrumTransformationModel" type="vwoc:PolynomialType"/>
      </sequence>
</complexType>

<complexType name="PolynomialType">
      <sequence>
            <element name="Monomial" type= "vwoc:MonomialType" maxOccurs ="unbounded" />
      </sequence>
</complexType>

<complexType name="MonomialType">
      <sequence>
            <element name="Variable" type= "vwoc:VariableType" minOccurs="0" maxOccurs
="unbounded" />
      </sequence>
      <attribute name="coefficient" type="double" use="optional"/>
</complexType>

<complexType name="VariableType">
     <attribute name="literal" type="string" use="required"/>
     <attribute name="exponent" type="positiveInteger" use="optional"/>
</complexType>

<complexType name="CosmeticCharacteristicType">
```

```
        <attribute name="form" type="vwoc:cosmeticFormType" use="optional"/>
        <attribute name="glossProperty" type="vwoc:glossPropertyType" use="optional"/>
        <attribute name="pearl" type="boolean" use="optional"/>
        <attribute name="transmittancy" type="double" use="optional"/>
</complexType>

<simpleType name="cosmeticFormType">
        <restriction base="string">
                <enumeration value ="Solid"/>
                <enumeration value ="Powder"/>
                <enumeration value ="Liquid"/>
                <enumeration value ="Cream"/>
                <enumeration value ="Gel"/>
        </restriction>
</simpleType>

<simpleType name="glossPropertyType">
        <restriction base="string">
                <enumeration value ="Glossy"/>
                <enumeration value ="Matt"/>
        </restriction>
</simpleType>
```

5.3.4.2 Binary Representation

CosmeticType {	***Number of bits***	***Mnemonic***
CharacteristicFlag	1	bslbf
CategoryFlag	1	bslbf
nameFlag	1	bslbf
brandFlag	1	bslbf
modelNumberFlag	1	bslbf
colorNumberFlag	1	bslbf
Color		CosmeticSpectrumType
if(CharacteristicFlag) {		
Characteristic		CosmeticCharacteristicType
}		
if(CategoryFlag) {		
Category	5	bslbf
}		
if(nameFlag) {		
name	See ISO 10646	UTF-8
}		

if(brandFlag) {		
brand	See ISO 10646	UTF-8
}		
if(modelNumberFlag) {		
modelNumber	See ISO 10646	UTF-8
}		
if(colorNumberFlag) {		
colorNumber	See ISO 10646	UTF-8
}		
}		
CosmeticSpectrumType {		
SpectraChoice	1	bslbf
if(SpectraChoice == 0) {		
heightSize		vluimsbf5
widthSize		vluimsbf5
dimensionSize	9	uimsbf
for(i=0;i<heightSize;i++) {		
for(k=0;k<widthSize;k++) {		
for(m=0;m<dimensionSize;m++) {		
Spectra[i][k][m]	32	fsfb
}		
}		
}		
}		
else {		
SpectraURI	See ISO 10646	UTF-8
}		
CosmeticSpectrumTransformationModel		PolynomialType

}		
PolynomialType {		
numOfMonomial		vluimsbf5
for(i=0;i<numOfMonomial;i++) {		
Monomial[i]		MonomialType
}		
}		
MonomialType {		
VariableFlag	1	bslbf
coefficientFlag	1	bslbf
if(VariableFlag) {		
numOfVariable		vluimsbf5
for(i=0;i<numOfVariable;i++) {		
Variable[i]		VariableType
}		
}		
if(coefficientFlag) {		
coefficient	32	fsfb
}		
}		
VariableType {		
exponentFlag	1	bslbf
literal	See ISO 10646	UTF-8
if(exponentFlag) {		
exponent		vluimsbf5
}		

}		
CosmeticCharacteristicType {		
formFlag	1	bslbf
glossPropertyFlag	1	bslbf
pearlFlag	1	bslbf
transmittancyFlag	1	bslbf
if(formFlag) {		
form	3	bslbf
}		
if(glossPropertyFlag) {		
glossProperty	1	Bslbf
}		
if(pearlFlag) {		
pearl	1	bslbf
}		
if(transmittancyFlag) {		
transmittancy	32	fsfb
}		
}		

5.3.4.3 Semantics

Name	*Definition*
`CosmeticType`	Tool for describing cosmetics.
`Color`	Describes color information of cosmetics.
`Characteristic`	Describes characteristic information of cosmetics.
`Category`	Describes kinds of cosmetics as a reference to a classification scheme term that shall be using the `mpeg7:termReferenceType` defined in ISO/IEC 15938-5:2003, 7.6. The CS that may be used for this purpose is the `CosmeticTypeCS` defined in A.2.5.1.

	<table><tr><th>*Name*</th><th>*Binary representation (5 bits)*</th><th>*Description*</th></tr><tr><td>Foundation</td><td>1</td><td>Foundation type cosmetics</td></tr><tr><td>Concealer</td><td>2</td><td>Concealer type cosmetics</td></tr><tr><td>Powder</td><td>3</td><td>Powder type cosmetics</td></tr><tr><td>Eyebrow</td><td>4</td><td>Eyebrow type cosmetics</td></tr><tr><td>EyeShadow</td><td>5</td><td>EyeShadow type cosmetics</td></tr><tr><td>Eyeliner</td><td>6</td><td>Eyeliner type cosmetics</td></tr><tr><td>Blusher</td><td>7</td><td>Blusher type cosmetics</td></tr><tr><td>Highlight</td><td>8</td><td>Highlight type cosmetics</td></tr><tr><td>Shading</td><td>9</td><td>Shading type cosmetics</td></tr><tr><td>Lipliner</td><td>10</td><td>Lipliner type cosmetics</td></tr><tr><td>Lipstick</td><td>11</td><td>Lipstick type cosmetics</td></tr><tr><td>Lipgloss</td><td>12</td><td>Lipgloss type cosmetics</td></tr><tr><td>Mascara</td><td>13</td><td>Mascara type cosmetics</td></tr><tr><td></td><td>0,14-31</td><td>Reserved</td></tr></table>
`Name`	Describes the name of cosmetics.
`Brand`	Describes the brand of cosmetics.
`modelNumber`	Describes the model number of cosmetics.
`colorNumber`	Describes the color number of cosmetics.
`Characteristic Flag`	This field, which is only present in the binary representation, signals the presence of Characteristic value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
`CategoryFlag`	This field, which is only present in the binary representation, signals the presence of Category value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
`nameFlag`	This field, which is only present in the binary representation, signals the presence of name value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.

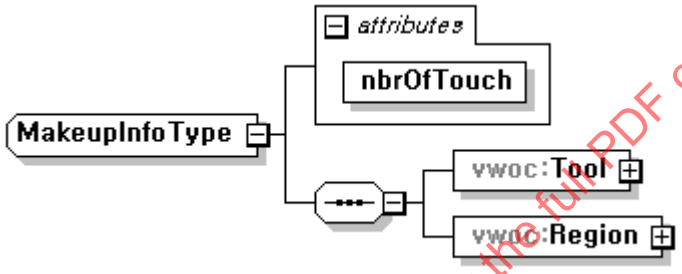
brandFlag	This field, which is only present in the binary representation, signals the presence of brand value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
modelNumberFlag	This field, which is only present in the binary representation, signals the presence of modelNumber value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
colorNumberFlag	This field, which is only present in the binary representation, signals the presence of colorNumber value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
CosmeticSpectrumType	Tool for describing color of cosmetics in the spectrum data. This type includes a spectrum data and a spectrum transformation model.
Spectra	Describes the spectrum data as a form of a matrix. The spectrum data is represented by an image size (i.e. a height and a width) and a spectrum dimension. The maximum size of the spectrum dimension is 301 since this covers a visible spectrum range between 400 nm and 700 nm for every 1 nm. The spectrum dimension can be subsampled linearly. For example, if the sampling rate is every 10 nm, the spectrum dimension becomes 31. Henceforth, the spectrum data can be represented by a three dimensional matrix that the size is *height × width × spectrum dimension*. A spectrum data (i.e. one pixel color) can be represented by a three dimension matrix with size of *1 × 1 × spectrum dimension*.
SpectraURI	Describes a URI that stores the spectrum data.
CosmeticSpectrumTransformationModel	A transformation model (e.g. equations) between a skin color spectrum and a cosmetic color spectrum.
SpectraChoice	This field, which is only present in the binary representation, describes which spectrum data shall be used. “0” means that the Spectra type shall be used, “1” means that the SpectraURI type shall be used
heightSize	Describes a height that the spectrum data.
widthSize	Describes a width that the spectrum data.
dimensionSize	Describes a dimension of spectrum that the spectrum data.
PolynomialType	Tool for describing a polynomial equation.
Monomial	Describes monomial equations, which constitute a polynomial equation.
numOfMonomial	This field, which is only present in the binary representation, specifies the number of `Monomial` information contained in the makeup information for avatar.
MonomialType	Tool for describing a monomial equation.
Variable	Describes variables of the monomial equation.

coefficient	Describes coefficients of the monomial equation. If the variable is not defined, this coefficient becomes a constant.
VariableFlag	This field, which is only present in the binary representation, signals the presence of Variable value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
coefficientFlag	This field, which is only present in the binary representation, signals the presence of coefficient value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
numOfVariable	This field, which is only present in the binary representation, specifies the number of Variable information contained in the makeup information for avatar.
VariableType	Tool for describing a variable.
exponent	Describes an exponent of the variable.
exponentFlag	This field, which is only present in the binary representation, signals the presence of exponent value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
literal	This field specifies the literal of Variable.
CosmeticCharacteristicType	Tool for describing the characteristic information of cosmetics.
form	Describes the forms of cosmetics.
glossProperty	Describe glossiness of cosmetics.
pearl	Describe the existence of pearl component.
transmittancy	Describe the transmittance of cosmetics.
formFlag	This field, which is only present in the binary representation, signals the presence of form value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
glossPropertyFlag	This field, which is only present in the binary representation, signals the presence of glossProperty value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
pearlFlag	This field, which is only present in the binary representation, signals the presence of pearl value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
transmittancyFlag	This field, which is only present in the binary representation, signals the presence of transmittancy value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
cosmeticFormType	Tool for describing the forms of cosmetics. The form of cosmetics is one of solid, powder, liquid, cream, and gel. In the binary description, the following mapping table is used. **Table 2 — cosmeticFormType** **cosmeticFormType** | **Sementics**

	000 solid 001 powder 010 liquid 011 cream 100 gel
glossPropertyType	Tool for describing the glossiness of cosmetics. The glossiness is either glossy or matt. In the binary description, the following mapping table is used. **Table 3 — glossPropertyType** **glossPropertyType** / **Sementics** 0 glossy 1 matt

5.3.5 MakeupInfoType

5.3.5.1 XML representation syntax

Diagram	
Source	<complexType name="MakeupInfoType"> <sequence> <element name="Tool" type="vwoc:MakeupToolType"/> <element name="Region" type="vwoc:MakeupRegionType"/> </sequence> <attribute name="nbrOfTouch" type="positiveInteger" use="required"/> </complexType>

5.3.5.2 Binary Representation

MakeupInfoType {	*Number of Bits*	*Mnemonic*
Tool		MakeupToolType
Region		MakeupRegionType
nbrOfTouch		vluimsbf5
}		

5.3.5.3 Semantics

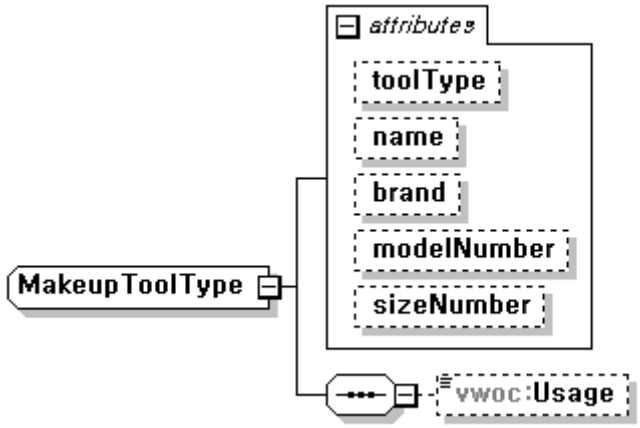
Name	*Definition*
MakeupInfoType	Tool for describing makeup information.
Tool	Describes makeup tools.

Region	Describes makeup regions of a face.
nbrOfTouch	Describes a number of makeup touches.

5.3.6 MakeupToolType

5.3.6.1 XML representation syntax

Diagram



Source

```
<complexType name="MakeupToolType">
    <sequence>
        <element name="Usage" type="mpeg7:termReferenceType" minOccurs="0"/>
    </sequence>
    <attribute name="toolType" type="vwoc:cosmeticToolType" use="optional"/>
    <attribute name="name" type="string" use="optional"/>
    <attribute name="brand" type="string" use="optional"/>
    <attribute name="modelNumber" type="string" use="optional"/>
    <attribute name="sizeNumber" type="string" use="optional"/>
</complexType>

<simpleType name="cosmeticToolType">
    <restriction base="string">
        <enumeration value="Brush"/>
        <enumeration value="Sponge"/>
        <enumeration value="PowderPuff"/>
        <enumeration value="Pen"/>
    </restriction>
</simpleType>
```

5.3.6.2 Binary representation

MakeupToolType {	*Number of Bits*	*Mnemonic*
UsageFlag	1	bslbf
toolTypeFlag	1	bslbf
nameFlag	1	bslbf
brandFlag	1	bslbf
modelNumberFlag	1	bslbf

sizeNumberFlag	1	bslbf
if(UsageFlag) {		
Usage	5	bslbf
}		
if(toolTypeFlag) {		
toolType	2	bslbf
}		
if(nameFlag) {		
name	See ISO 10646	UTF-8
}		
if(brandFlag) {		
brand	See ISO 10646	UTF-8
}		
if(modelNumberFlag) {		
modelNumber	See ISO 10646	UTF-8
}		
if(sizeNumberFlag) {		
sizeNumber	See ISO 10646	UTF-8
}		
}		

5.3.6.3 Semantics

Name	*Definition*
`MakeupToolType`	Tool for describing makeup tool information.
`Usage`	Describe the purpose of a makeup tool as a reference to a classification scheme term that shall be using the `mpeg7:termReferenceType` defined in ISO/IEC 15938-5:2003, 7.6. The CS that may be used for this purpose is the `CosmeticTypeCS` defined in A.2.5.1.

Name	*Binary representation (5 bits)*	*Description*
Foundation	1	Usage for foundation

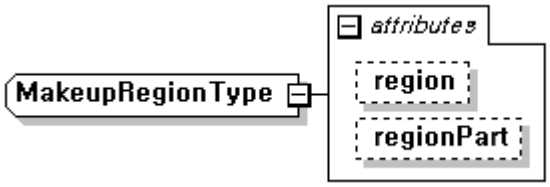
	<table><tr><td>Concealer</td><td>2</td><td>Usage for concealer</td></tr><tr><td>Powder</td><td>3</td><td>Usage for powder</td></tr><tr><td>Eyebrow</td><td>4</td><td>Usage for eyebrow</td></tr><tr><td>EyeShadow</td><td>5</td><td>Usage for eyeShadow</td></tr><tr><td>Eyeliner</td><td>6</td><td>Usage for eyeliner</td></tr><tr><td>Blusher</td><td>7</td><td>Usage for blusher</td></tr><tr><td>Highlight</td><td>8</td><td>Usage for highlight</td></tr><tr><td>Shading</td><td>9</td><td>Usage for shading</td></tr><tr><td>Lipliner</td><td>10</td><td>Usage for lipliner</td></tr><tr><td>Lipstick</td><td>11</td><td>Usage for lipstick</td></tr><tr><td>Lipgloss</td><td>12</td><td>Usage for lipgloss</td></tr><tr><td>Mascara</td><td>13</td><td>Usage for mascara</td></tr><tr><td></td><td>0,14-31</td><td>Reserved</td></tr></table>
`toolType`	Describe the type of a makeup tool.
`name`	Describe the name of a makeup tool.
`brand`	Describe the brand of a makeup tool.
`modelNumber`	Describe the model number of a makeup tool.
`sizeNumber`	Describe the size number of a makeup tool.
`cosmeticToolType`	Tool for describing the type of a makeup tool. The type is one of brush, sponge, powder puff, and pen. In the binary description, the following mapping table is used. **Table 4 — cosmeticToolType** <table><tr><th>cosmeticToolType</th><th>Sementics</th></tr><tr><td>00</td><td>brush</td></tr><tr><td>01</td><td>sponge</td></tr><tr><td>10</td><td>powder puff</td></tr><tr><td>11</td><td>pen</td></tr></table>
`UsageFlag`	This field, which is only present in the binary representation, signals the presence of `Usage` value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
`toolTypeFlag`	This field, which is only present in the binary representation, signals the presence of name `toolType` attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
`nameFlag`	This field, which is only present in the binary representation, signals the presence of `name` value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.

brandFlag	This field, which is only present in the binary representation, signals the presence of brand value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
modelNumberFlag	This field, which is only present in the binary representation, signals the presence of modelNumber value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.
sizeNumberFlag	This field, which is only present in the binary representation, signals the presence of sizeNumber value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.

5.3.7 MakeupRegionType

5.3.7.1 XML representation syntax

Diagram



Source

```
<complexType name="MakeupRegionType">
    <attribute name="region" type="vwoc:makeupRegionType" use="optional"/>
    <attribute name="regionNumber" type="vwoc:unsigned4Vector" use="optional"/>
</complexType>

<simpleType name="makeupRegionType">
    <restriction base="string">
        <enumeration value ="All"/>
        <enumeration value ="Lips"/>
        <enumeration value ="Eyebrow"/>
        <enumeration value ="Periocular"/>
        <enumeration value ="Cheek"/>
        <enumeration value ="Highlight"/>
        <enumeration value ="Shadow"/>
        <enumeration value ="Shading"/>
    </restriction>
</simpleType>

<simpleType name="unsigned4Vector">
    <list itemType="mpeg7:unsigned4"/>
</simpleType>
```

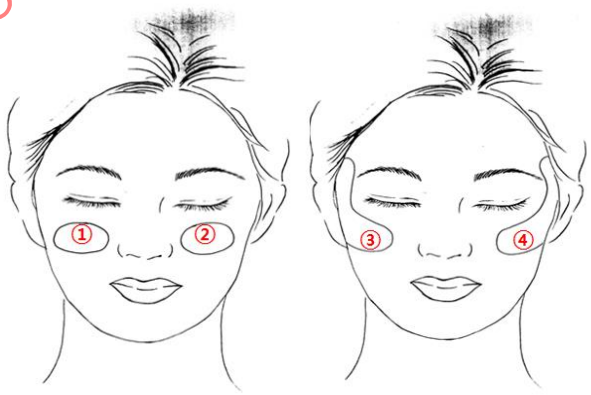
5.3.7.2 Binary representation

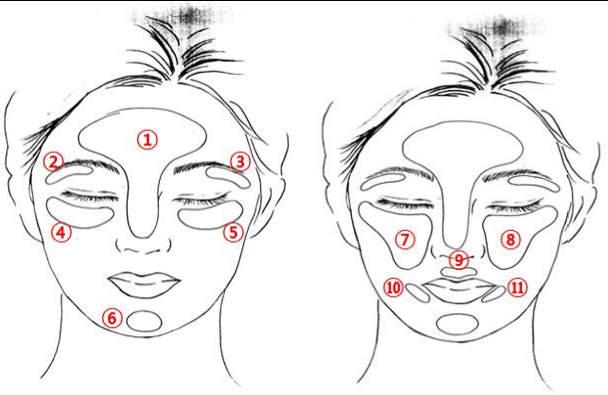
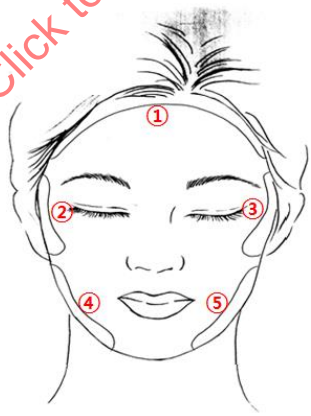
MakeupRegionType {	*Number of Bits*	*Mnemonic*
regionFlag	1	bslbf
regionNumberFlag	1	bslbf
if(regionFlag) {		
Region	4	bslbf
}		

if(regionNumberFlag) {		
numOfRegionNumber		vluimsbf5
for(k=0;k<numOfRegionNumber;k++) {		
regionNumber[k]	4	uimsbf
}		
}		
}		

5.3.7.3 Semantics

Name	*Definition*
`MakeupRegionType`	Tool for describing the region information of a makeup.
`Region`	Describes the region of a makeup.
`regionNumber`	Specifies the detailed location of a makeup in a region. The detailed location is represented by a number as explained in the following table. Multiple region parts can be assigned simultaneously.

All	0: all
Lips	0: all, 1: upper lip, 2: lower lip
Eyebrow	0: all, 1: right eyebrow, 2: left eyebrow
Periocular	0: all, 1: right periocular, 2: left periocular
Cheek	0: all, 
Highlight	0: all,

			
	Shadow	0: all, 	
	Shading	0: all, 	
makeupRegionType	Tool for describing the makeup regions. The makeup region is one of all, lips, eyebrow, periocular, cheek, highlight, shadow, and shading. In the binary description, the following mapping table is used. **Table 5 — makeupRegionType** **makeupRegionType** / **Sementics** 0000 / all 0001 / lips 0010 / eyebrow 0011 / periocular 0100 / cheek		

	0101	highlight
	0110	shadow
	0111	shading
	1000 - 1111	reserved
regionFlag	This field, which is only present in the binary representation, signals the presence of region value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.	
regionNumberFlag	This field, which is only present in the binary representation, signals the presence of regionNumber value attribute. A value of “1” means the attribute shall be used and “0” means the attribute shall not be used.	
numOfRegionNumber	This field, which is only present in the binary representation, specifies the number of regionNumber attribute values.	

5.3.8 Examples

This example shows the description of makeup virtual object with the following semantics. The list of avatar contains one makeup avatar. The makeup avatar has the gender “female”, the id “MakeupAvatar001”, and information of cosmtics and makeup. The costemtics has the name “Prorance Finish Water Gloe-skinfinish”, the brand “Prorance”, the color number “M11 Shine Pink”, the model number “P24036587”, and information of the cosmetic colors and their characteristics. The cosmetic colors have the spectra URI “http://www.etri.re.kr/makeupAvatar/cosmeticSpectrum/sample003.spt” and the cosmetic spectrum transformation model. The cosmetic spectrum transformation model is a polynomial equation of “$2{,}1x^2 + 1{,}5xy^2 + 3{,}7$”. The cosmetic characteristics have a “Solid” form, a pearl, glossy, and a transmittancy value of “0,1”. The makeup method has two of the number of make touch and includes information of the makeup tools and the makeup regions. The makeup tool has a name “Bobbi Brown Blush Brush” and a brand “Bobbi Brown”, a tool type “Brush”, the size number “M2”, the model number “B02351269”, and the usage of “urn:mpeg:mpeg-v:01-VWOC-CosmeticCategoryCS-NS:Blusher”. The makeup region is “Cheek” and the region numbers in the “Cheek” are one and two.

```
<vwoc:VWOCInfo>
  <vwoc:AvatarList>
    <vwoc:Avatar xsi:type="vwoc:MakeupAvatarType" gender="female"
id="MakeupAvatar001">
     <vwoc:Cosmetic name="Prorance Finish Water Gloe-skinfinish" brand="Prorance"
colorNumber="M11 Shine Pink" modelNumber="P24036587">
       <vwoc:Color>
         <vwoc:SpectraURI>
           http://www.etri.re.kr/makeupAvatar/cosmeticSpectrum/sample003.spt
         </vwoc:SpectraURI>
         <!-- polynomial => 2.1x^2 + 1.5xy^2 + 3.7 -->
         <vwoc:CosmeticSpectrumTransformationModel>
           <vwoc:Monomial coefficient="2.1">
             <vwoc:Variable literal="x" exponent="2"/>
           </vwoc:Monomial>
           <vwoc:Monomial coefficient="1.5">
             <vwoc:Variable literal="x" exponent="1"/>
             <vwoc:Variable literal="y" exponent="2"/>
           </vwoc:Monomial>
           <vwoc:Monomial coefficient="3.7"/>
         </vwoc:CosmeticSpectrumTransformationModel>
       </vwoc:Color>
       <vwoc:Characteristic form="Solid" pearl="true" glossProperty="Glossy"
transmittancy="0.1"/>
```



```
      <vwoc:Category>urn:mpeg:mpeg-v:01-VWOC-CosmeticCategoryCS-
NS:Blusher</vwoc:Category>
    </vwoc:Cosmetic>
    <vwoc:MakeupInfo nbrOfTouch="2">
      <vwoc:Tool name="Bobbi Brown Blush Brush" brand="Bobbi Brown"
toolType="Brush" sizeNumber="M2" modelNumber="B02351269">
       <vwoc:Usage>urn:mpeg:mpeg-v:01-VWOC-CosmeticCategoryCS-
NS:Blusher</vwoc:Usage>
      </vwoc:Tool>
      <vwoc:Region region="Cheek" regionNumber="1 2"/>
    </vwoc:MakeupInfo>
  </vwoc:Avatar>
 </vwoc:AvatarList>
</vwoc:VWOCInfo>
```

6 Virtual object metadata

6.1 General

Virtual object metadata as a (visual) representation of virtual objects inside the environment serves the following purposes:

- characterize various kinds of objects within the VE;
- provide an interaction between virtual object and avatar;
- provide an interaction with the VE.

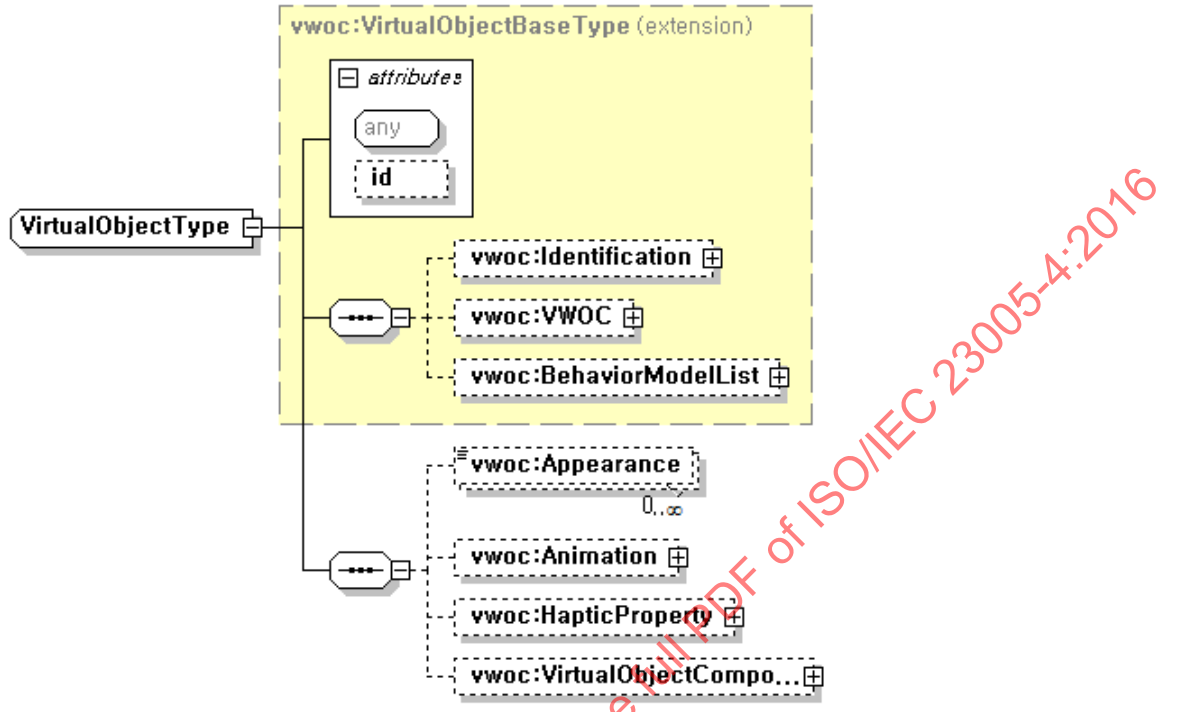
The "virtual object" element is composed of following type of data with the extension of the base type of a virtual object.

- **Appearance**: contains the high level description of the appearance and may refer a media containing the exact geometry, texture and haptic properties.
- **Animation**: contains the description of a set of animation sequences that the object is able to perform and may refer to several medias containing the exact (geometric transformations and deformations) animation parameters.
- **HapticProperty**: contains the description of the haptic property of the virtual object.
- **Virtual object components**: contains the list of the virtual objects which are concatenated to the virtual object as components.

6.2 VirtualObjectType

6.2.1 XML representation syntax

Diagram	



Source

```
<complexType name="VirtualObjectType">
    <complexContent>
        <extension base="vwoc:VirtualObjectBaseType">
            <sequence>
                <element name="Appearance" type="anyURI" minOccurs="0" maxOccurs="unbounded"/>
                <element name="Animation" type="vwoc:VOAnimationType" minOccurs="0"/>
                <element name="HapticProperty" type="vwoc:VWOHapticPropertyType" minOccurs="0"/>
                <element name="VirtualObjectComponents" type="vwoc:VirtualObjectListType"
minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```


6.2.2 Binary representation syntax

VirtualObjectType{	***Number of bits***	***Mnemonic***
AppearanceFlag	1	bslbf
AnimationFlag	1	bslbf
HapticPropertyFlag	1	bslbf
VirtualObjectComponentsFlag	1	bslbf
VirtualObjectBase		VirtualObjectBaseType

if(AppearanceFlag){		
NumAppearance		vluimsbf5
for(k=0; k< NumAppearance; k++){		
Appearance[k]	See ISO 10646	UTF-8
}		
}		
if(AnimationFlag){		
Animation		VOAnimationType
}		
if(HapticPropertyFlag){		
HapticProperty		VWOHapticPropertyType
}		
if(VirtualObjectComponentsFlag){		
VirtualObjectComponents		VirtualObjectListType
}		
}		

6.2.3 Semantics

Name	*Definition*
`VirtualObjectType`	A type that provides a representation of virtual object inside the environment.
`AppearanceFlag`	This field, which is only present in the binary representation, signals the presence of the `Appearance` elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.
`AnimationFlag`	This field, which is only present in the binary representation, signals the presence of the `Animation` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`HapticPropertyFlag`	This field, which is only present in the binary representation, signals the presence of the `HapticProperty` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`VirtualObjectComponentsFlag`	This field, which is only present in the binary representation, signals the presence of the `VirtualObjectComponents` element. ”1” means that the element shall be used. ”0” means that the element shall not be used.
`VirtualObjectBase`	Contains the base type defined by `VirtualObjectBaseType`.
`NumAppearance`	This field, which is only present in the binary representation, specifies the number of virtual object appearance elements.
`Appearance`	This element contains one or more resource link(s) to appearance(s) file(s) describing the visual and tactile elements of the object.

Animation	This element contains a set of metadata describing pre-recorded animations associated with the object.
HapticProperty	This element contains a set of high level descriptors of the haptic properties defined in the VWOHapticPropertyType of the virtual world object.
VirtualObject Components	This element contains the list of the virtual objects which are concatenated to the virtual object as components.

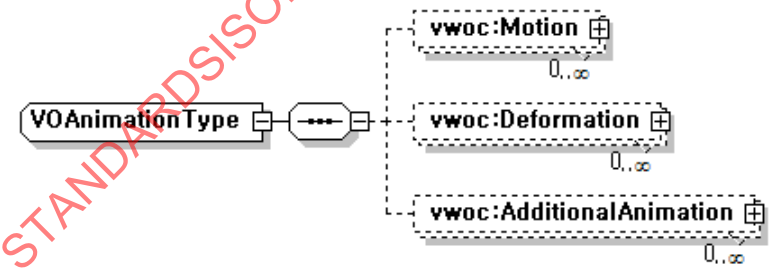
6.2.4 Examples

This example shows the description of virtual object information with the following semantics. The list of virtual objects that contains 2 virtual objects is given. One virtual object whose id is "virtualObject_001" has the identification name as "clothe" and the appearance resource of "http://clothsdb.com/clothe_001.clo." The other virtual object whose id is "virtualObject_002" has the appearance resource of "http://3DmodelDb.com/object_0001.3ds" and the animation, the name of which is "Turn360" and the resource of which is "http://voAnimationdb.com/turn_360.bvh."

```
<vwoc:VirtualObjectList>
  <vwoc:VirtualObject xsi:type="vwoc:VirtualObjectType" id="virtualObject_001">
    <vwoc:Identification name="clothe"/>
    <vwoc:Appearance>http://clothsdb.com/clothe_001.clo</vwoc:Appearance>
  </vwoc:VirtualObject>
  <vwoc:VirtualObject xsi:type="vwoc:VirtualObjectType" id="virtualObject_002">
    <vwoc:Appearance>http://3DmodelDb.com/object_0001.3ds</vwoc:Appearance>
    <vwoc:Animation>
      <vwoc:Motion>
        <vwoc:Name>urn:mpeg:mpeg-v:01-VWOC-VOMotionCS-NS:turn360</vwoc:Name>
        <vwoc:Uri>http://voAnimationdb.com/turn_360.bvh</vwoc:Uri>
      </vwoc:Motion>
    </vwoc:Animation>
  </vwoc:VirtualObject>
</vwoc:VirtualObjectList>
```

6.2.5 VOAnimationType

6.2.5.1 XML representation syntax

Diagram	
Source	<complexType name="VOAnimationType"> <sequence> <element name="Motion" type="vwoc:AnimationDescriptionType" minOccurs="0" maxOccurs="unbounded"/> <element name="Deformation" type="vwoc:AnimationDescriptionType" minOccurs="0" maxOccurs="unbounded"/> <element name="AdditionalAnimation" type="vwoc:AnimationResourcesDescriptionType" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType>

6.2.5.2 Binary representation syntax

VOAnimationType{	***Number of bits***	***Mnemonic***
MotionFlag	1	bslbf
DeformationFlag	1	bslbf
AdditionalAnimationFlag	1	bslbf
if(MotionFlag){		
NumMotion		vluimsbf5
for(k=0;k<NumMotion;k++){		
Motion[k]		AnimationDescriptionType
}		
}		
if(Deformation Flag){		
NumDeformation		vluimsbf5
for(k=0;k< NumDeformation;k++){		
Deformation[k]		AnimationDescriptionType
}		
}		
if(AdditionalAnimation Flag){		
NumAdditionalAnimation		vluimsbf5
for(k=0;k< NumAdditionalAnimation;k++){		
AdditionalAnimation[k]		AnimationResourcesDescriptionType
}		
}		
}		

6.2.5.3 Semantics

Name	*Definition*

<table>
<tr><td>VOAnimationType</td><td>A type that contains the description of an animation and may refer several media containing the exact animation parameters.</td></tr>
<tr><td>MotionFlag</td><td>This field, which is only present in the binary representation, signals the presence of the Motion elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.</td></tr>
<tr><td>DeformationFlag</td><td>This field, which is only present in the binary representation, signals the presence of the Deformation elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.</td></tr>
<tr><td>AdditionalAnimationFlag</td><td>This field, which is only present in the binary representation, signals the presence of the AdditionalAnimation elements. ”1” means that the elements shall be used. ”0” means that the elements shall not be used.</td></tr>
<tr><td>NumMotion</td><td>This field, which is only present in the binary representation, specifies the number of rigid motion elements.</td></tr>
<tr><td>Motion</td><td>Set of animations defined as a rigid motion:
Examples of motion animations defined in VOMotionCS in A.2.4.2 are as follows.
<table>
<tr><th>Name</th><th>Binary representation (6 bits)</th><th>description</th></tr>
<tr><td>moveDown</td><td>1</td><td>move down</td></tr>
<tr><td>moveLeft</td><td>2</td><td>move left</td></tr>
<tr><td>moveRight</td><td>3</td><td>move right</td></tr>
<tr><td>moveUp</td><td>4</td><td>move up</td></tr>
<tr><td>turn180</td><td>5</td><td>turn 180</td></tr>
<tr><td>turnback180</td><td>6</td><td>turn back 180</td></tr>
<tr><td>turnLeft</td><td>7</td><td>turn left</td></tr>
<tr><td>turnRight</td><td>8</td><td>turn right</td></tr>
<tr><td>turn360</td><td>9</td><td>turn 360</td></tr>
<tr><td>turnback360</td><td>10</td><td>turn back 360</td></tr>
<tr><td>freeDirection</td><td>11</td><td>free direction</td></tr>
<tr><td>appear</td><td>12</td><td>appear</td></tr>
<tr><td>away</td><td>13</td><td>away</td></tr>
<tr><td>disappear</td><td>14</td><td>disappear</td></tr>
<tr><td>falldown</td><td>15</td><td>fall down</td></tr>
<tr><td>bounce</td><td>16</td><td>bounce</td></tr>
<tr><td>toss</td><td>17</td><td>toss</td></tr>
<tr><td>spin</td><td>18</td><td>spin</td></tr>
<tr><td>fly</td><td>19</td><td>fly</td></tr>
<tr><td>vibrate</td><td>20</td><td>vibrate</td></tr>
<tr><td>flow</td><td>21</td><td>flow</td></tr>
<tr><td></td><td>0,22-64</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>NumDeformation</td><td>This field, which is only present in the binary representation, specifies the number of deformation action elements.</td></tr>
<tr><td>Deformation</td><td>Set of deformation animations.
Examples of deformation animations defined in VODeformationCS in A.2.4.1 are as follows.
<table>
<tr><th>Name</th><th>Binary representation (6 bits)</th><th>description</th></tr>
<tr><td>flip</td><td>1</td><td>flip</td></tr>
<tr><td>stretch</td><td>2</td><td>stretch</td></tr>
<tr><td>swirl</td><td>3</td><td>swirl</td></tr>
<tr><td>twist</td><td>4</td><td>twist</td></tr>
<tr><td>bend</td><td>5</td><td>bend</td></tr>
<tr><td>roll</td><td>6</td><td>roll</td></tr>
<tr><td>press</td><td>7</td><td>press</td></tr>
<tr><td>fallToPieces</td><td>8</td><td>fall to pieces</td></tr>
<tr><td>explode</td><td>9</td><td>explode</td></tr>
</table></td></tr>
</table>

	fire	10	fire	
		0,11-64	Reserved	
NumAdditionalAnimation	This field, which is only present in the binary representation, specifies the number of additional animation elements.			
AdditionalAnimation	Element that contains, if exist, one or more link(s) to animation(s) file(s).			

6.2.5.4 Examples

This example shows the description of object animation information with the following semantics. Among all animations, motion type animation of turning 360° is given. The animation resource is saved at "http://voAnimationdb.com/turn_360.bvh" and the value of animationID, its identifier is "Animation3." The intensity shall be played once with duration of 30 seconds.

```
<vwoc:Animation>
   <vwoc:Motion animationID="Animation3" duration="30" loop="1">
      <vwoc:Name> urn:mpeg:mpeg-v:01-VWOC-VOMotionCS-NS:turn360</vwoc:Name>
      <vwoc:Uri>http://voAnimationdb.com/turn_360.bvh</vwoc:Uri>
   </vwoc:Motion>
</vwoc:Animation>
```

Annex A
(normative)

Classification Schemes

A.1 General

This Annex specifies a set of classification schemes that may be used by applications using description tools specified in this part of ISO/IEC 23005. Applications need not use these classification schemes; they can use proprietary or third party ones. However, if they choose to use the classification schemes defined in this Annex, no modifications or extensions are allowed to these classification schemes. The classification schemes in this Annex are specified using the `ClassificationScheme` defined in ISO/IEC 15938-5:2003, 7.6. All of the classification schemes defined in this Annex are uniquely identified by a URN following the "`urn:mpeg:mpeg-v:01-VWOC-NameCS-NS`" namespace identifier where `Name` should be replaced with the name of the classification scheme. For example, the URN "`urn:mpeg:mpeg-v:01-VWOC-IdleAnimationCS-NS`" identifies the classification scheme provided for `Idle` animation types of AvatarAnimationType.

In some cases, there are several classification schemes associated with a single description. In such a case, any one of these classification schemes (as well as classification schemes not defined in this specification) may be used depending on the application domain.

A.2 Classification schemes

A.2.1 Classification scheme for input events

A.2.1.1 MouseEventCS

This subclause contains a classification scheme for mouse events. `MouseEventCS` corresponds to the `Mouse` element value in `VWOEventType`.

```
<ClassificationScheme uri="urn:mpeg:mpeg-v:01-VWOC-MouseEventCS-NS">
    <Term termID="click">
        <Name xml:lang="en">Click</Name>
        <Definition xml:lang="en">
            Describes the event of click the left button of a mouse(Tap swiftly.)
       </Definition>
    </Term>
    <Term termID="doubleclick">
        <Name xml:lang="en">DoubleClick</Name>
        <Definition xml:lang="en">
            Describes the event of double-click the left button of a mouse(Tap
swiftly and with the taps as close to each other as possible).
       </Definition>
    </Term>
    <Term termID="leftbtndown">
        <Name xml:lang="en">LeftButtonDown</Name>
        <Definition xml:lang="en">
            Describes the event which takes place at the moment of holding down the
left button of a mouse.
       </Definition>
    </Term>
    <Term termID="leftbtnup">
        <Name xml:lang="en">LeftButtonUP</Name>
```



```
        <Definition xml:lang="en">
            Describes the event which takes place at the moment of releasing the
left button of a mouse.
        </Definition>
    </Term>
    <Term termID="rightbtndown">
        <Name xml:lang="en">RightButtonDown</Name>
        <Definition xml:lang="en">
            Describes the event which takes place at the moment of holding down the
right button of a mouse.
        </Definition>
    </Term>
    <Term termID="rightbtnup">
        <Name xml:lang="en">RightButtonUP</Name>
        <Definition xml:lang="en">
            Describes the event which takes place at the moment of releasing the
right button of a mouse.
        </Definition>
    </Term>
    <Term termID="wheelbtndown">
        <Name xml:lang="en">WheelButtonDown</Name>
        <Definition xml:lang="en">
            Describes the event which takes place at the moment of pushing the
wheel button of a mouse.
        </Definition>
    </Term>
    <Term termID="wheelbtnup">
        <Name xml:lang="en">WheelButtonUp</Name>
        <Definition xml:lang="en">
            Describes the event which takes place at the moment of releasing the
wheel button of a mouse.
        </Definition>
    </Term>
    <Term termID="wheelscrolldown">
        <Name xml:lang="en">WheelScrollDown</Name>
        <Definition xml:lang="en">
            Describes the mouse event which takes place at the moment of scrolling
the wheel down.
        </Definition>
    </Term>
    <Term termID="wheelscrollup">
        <Name xml:lang="en">WheelScrollUp</Name>
        <Definition xml:lang="en">
            Describes the mouse event which takes place at the moment of scrolling
the wheel up.
        </Definition>
    </Term>
    <Term termID="move">
        <Name xml:lang="en">Move</Name>
        <Definition xml:lang="en">
            Describes the event which takes place while changing the mouse
position.
        </Definition>
    </Term>
</ClassificationScheme>
```

A.2.2 Classification scheme for hairstyles

This subclause contains a classification scheme for hairstyles. `HairStyleCS` corresponds to the `HairStyle` element value in `HairType`.

```
<ClassificationScheme uri="urn:mpeg:mpeg-v:01-VWOC-HairStyleCS-NS">
    <Term termID="afro">
        <Name xml:lang="en">Afro</Name>
        <Definition xml:lang="en">
            Describes the style of the afro hair.
       </Definition>
    </Term>
    <Term termID="bun">
        <Name xml:lang="en">Bun</Name>
        <Definition xml:lang="en">
            Describes the style of the bun hair.
       </Definition>
    </Term>
    <Term termID="combover">
        <Name xml:lang="en">Combover</Name>
        <Definition xml:lang="en">
            Describes the style of the combover hair.
       </Definition>
    </Term>
    <Term termID="crewcut">
        <Name xml:lang="en">Crewcut</Name>
        <Definition xml:lang="en">
            Describes the style of the crewcut hair.
       </Definition>
    </Term>
    <Term termID="mohawk">
        <Name xml:lang="en">Mohawk</Name>
        <Definition xml:lang="en">
            Describes the style of the mohawk hair.
       </Definition>
    </Term>
    <Term termID="odando">
        <Name xml:lang="en">Odando</Name>
        <Definition xml:lang="en">
            Describes the style of the odando hair.
       </Definition>
    </Term>
    <Term termID="pigtails">
        <Name xml:lang="en">Pigtails</Name>
        <Definition xml:lang="en">
            Describes the style of the pigtails hair.
       </Definition>
    </Term>
    <Term termID="pompadour">
        <Name xml:lang="en">Pompadour</Name>
        <Definition xml:lang="en">
            Describes the style of the pompadour hair.
       </Definition>
    </Term>
```