INTERNATIONAL STANDARD

**ISO/IEC 23007-2**

First edition
2012-10-01

# Information technology — Rich media user interfaces —

## Part 2:
## Advanced user interaction (AUI) interfaces

*Technologies de l'information — Interfaces d'utilisateur au support riche —*

*Partie 2: Interfaces d'intéraction d'utilisateur avancé (AUI)*

Reference number
ISO/IEC 23007-2:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

## Contents

# Foreword

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

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

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

ISO/IEC 23007-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 23007 consists of the following parts, under the general title *Information technology — Rich media user interfaces*:

- *Part 1: Widgets*
- *Part 2: Advanced user interaction (AUI) interfaces*
- *Part 3: Conformance and reference software*

# Introduction

R&D activities regarding advanced user interaction devices and mechanisms (often referred as HCI) such as motion and voice recognitions were used to reside within academia and lab environments. However, user interaction devices of industry have evolved dramatically in recent years with the maturing technologies. Consequently various advanced interaction devices, such as multi-touch pad, g-sensor, etc., are already incorporated into consumer electronics and offered in the market.

Although multimedia technologies have been studied to provide mature applicable technologies; however, current user interaction standards mostly focus on basic interaction devices such as pointing and keying devices. This lack of support results in unavailability to utilize such industry ready advanced interaction devices in a standard way.

This part of ISO/IEC 23007 presents the list of new data formats for scene specific advanced user interaction devices. It also includes the high-level view of the relationship between semantically recognized information and physical sensed information. Moreover it provides other interface formats of advanced user interaction devices in order to show the applicability of this International Standard with the existing standard.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this part of ISO/IEC 23007 may involve the use of patents.

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

The holder of this patent right has assured the ISO and IEC that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with the ISO and IEC. Information may be obtained from the companies listed in Annex D.

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

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

# Information technology — Rich media user interfaces —

# Part 2:
Advanced user interaction (AUI) interfaces

## 1 Scope

This part of ISO/IEC 23007 specifies advanced user interaction interfaces (AUI) to support various advanced user interaction devices. The AUI interface is a part of the bridge between scene descriptions and system resources. A scene description is a self-contained living entity composed of video, audio, 2D graphics objects, and animations. Through the AUI interfaces or other existing interfaces such as DOM events, a scene description accesses interesting system resources to interact with users. In general, a scene composition is conducted by a third party and remotely deployed.

## 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.

W3C DOM Events, "*Document Object Model Events*", available at http://www.w3.org/TR/DOM-Level-2-Events/events.html

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

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

## 3 Terms, definitions, and abbreviated terms

### 3.1 Terms and definitions

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

**3.1.1**
**advanced user interaction**
interactions that utilize more advanced interaction devices

EXAMPLE Multi-pointing user interaction, Motion sensing, Gaze tracking

**3.1.2**
**sensor**
combination of hardware and software or just an instance of software that detects features from real world

**3.1.3**
**presentation engine**
entity processing the Scene Description to provide its animated and interactive behaviour through composition and rendering

**3.1.4**
**scene description**
description defining an audiovisual and graphics presentation in terms of spatiotemporal layout, and interactions by using text, graphics, animations, images, videos, sounds, etc

**3.1.5**
**widget**
self-contained entity, with extensive communication capabilities, within a Rich Media User Interface; composed of a Manifest and associated Resources, including Scene Descriptions for the Full and Simplified Representations and Context Information

### 3.2 Abbreviated terms

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

| | |
|---|---|
| BIFS | BInary Format for Scene |
| LASeR | Lightweight Application Scene Representation. |
| SVG | Scalable Vector Graphics |
| XML | eXtensible Markup Language |
| AUI | Advanced User Interaction |
| MPEG-U | ISO/IEC 23007 |
| MPEG-V | ISO/IEC 23005 |

## 4 Overview

### 4.1 Introduction

This Subclause introduces the AUI patterns which this part of ISO/IEC 23007 provides as below.

1. The geometric patterns are a set of geometric shapes which are recognized with sensed geometric information as 2D or 3D Cartesian positions. Currently, the standard is trying to define the following geometric interactivity patterns: Point, Line, Rect., Arc and Circle patterns.
2. Instead of speaking or writing a word, simple well-known gestures help to communicate with others. For instance, V sign and Rock sign, which are well-known semantics, are already used in various situations. Therefore, the standard provides a container format for symbolic patterns and a classification scheme to enlist well-known symbolic shapes.
3. Many applications adopt well-known touch patterns for users to control themselves. In this part of ISO/IEC 23007, a container format for well-known touch patterns and a classification scheme to enlist basic touch patterns are provided.
4. This part of ISO/IEC 23007 describes the hand posture and gesture patterns to support the intuitive hand based interaction for scene description. For example, if a user wants to control an object in a scene description, the user in the real life makes a hand gesture such as grabbing, fist, and open palm which would be a good candidate gesture to support such an interaction modality. In addition, when the user wants to interact with people using hand gestures, “waving a hand” gesture is common semantic to all the human beings, even though the contextual meaning of the gesture may vary.

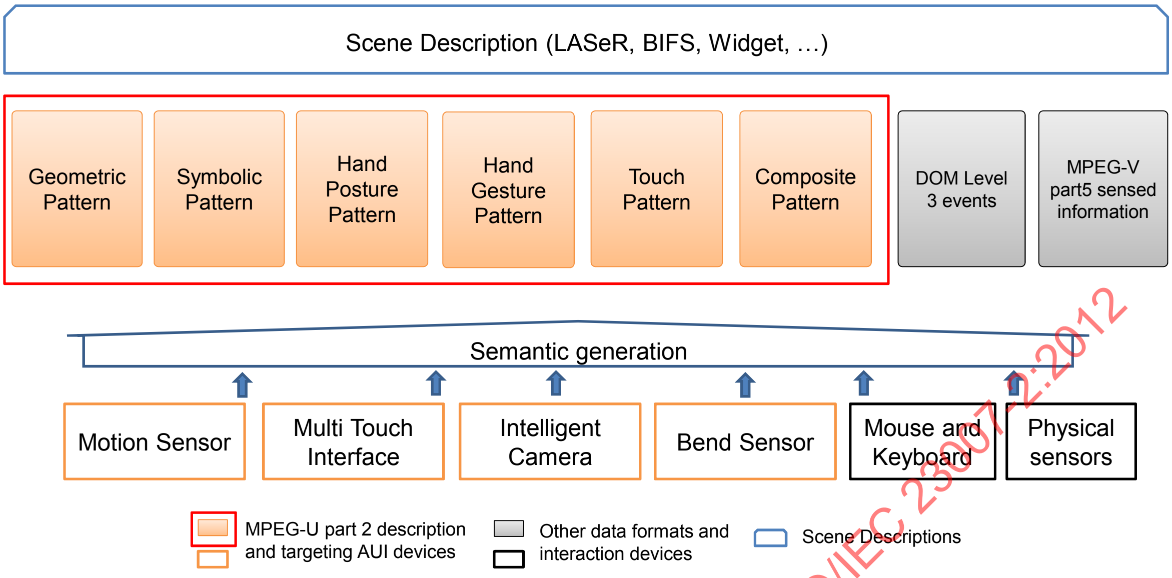

**Figure 1 — Architecture of MPEG-U part 2**

Figure 1 presents the overall architecture of this part of ISO/IEC 23007. User interaction devices such as motion sensors and multi touch interfaces generate the physical sensed information from user's environment. According to the use cases which are mentioned in the context and objectives document for this part of ISO/IEC 23007, a set of physical information can generate a semantic meaning by a recognition process. For instance, some feature points drawn by user's finger can be understood as a circle which is specified with a circle center position and a radius value. Therefore, this part of ISO/IEC 23007 provides a set of data formats which describes geometric patterns, symbolic patterns, touch patterns, hand posture patterns and their composite pattern.

At the right side of the above figure, other standard activities are also providing similar data structures for new user interaction interfaces such as physical sensors and even new features of mouse and keyboard. Therefore, scene description can use not only the data formats of this part of ISO/IEC 23007 and existing data sets from others.

## 4.2 Advanced user interaction devices

Recently various new types of user interaction devices are available on the market such as multi-touch pad, motion sensing remote controller, etc. Figure 2 shows a few available advanced user interaction devices and Figure 3 shows multi-touch devices currently available on the market.

**Figure 2 — Motion sensing remote controllers**

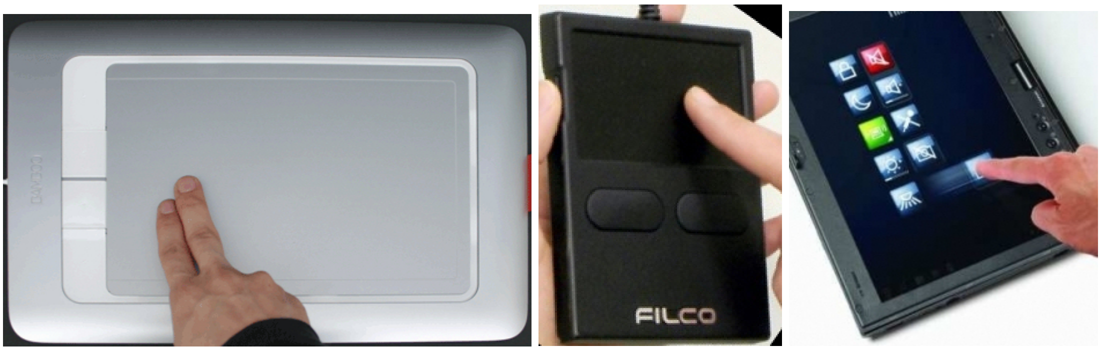

Figure 3 — Multi-touch devices

Among various possible Advanced User Interaction (AUI) Interfaces, multi-touch interface is the first recognized interaction interface to be ready both in terms of device availability on the market and in terms of applicability to various MPEG scene representation standards.

Almost unlimited number of actions can be mapped with the combination of 1) the number of fingers touched, 2) the moving direction of each touched finger, 3) the amount of displacement of each finger, etc. Therefore, multi-touch interface should be defined by carefully extracting necessary core attributes and make the structure expandable.

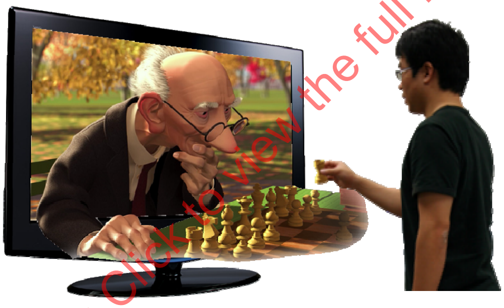

Figure 4 — Hand based 3D interaction

Another example among possible Advanced User Interaction (AUI) Interfaces is related to hand based interface shown in Figure 4. If a user wants to select an object and move it to a certain place in the scene, the user in the real life moves his/her hand toward the object, and then move the object while holding and grabbing the object. The user finally releases the object by opening the hand. That is, hand postures such as grabbing, fist, open palm, pointing, thumb-up, and thumb-down would be a good candidate posture to support such an interaction modality. The semantics of the postures shall be selected as the descriptions of the shapes, which are independent upon the cultural difference.

## 5 Interactivity patterns

### 5.1 Common types

This Clause describes common types including the schema wrapper conventions, basic data types. The types defined in this Clause are defined to be used in this part of ISO/IEC 23007, and are not intended to be instantiated again.

### 5.1.1 Schema wrapper conventions

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

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:aui="urn:mpeg:mpegu:schema:aui:2012"
   xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
   targetNamespace="urn:mpeg:mpegu:schema:aui:2012"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified" version="ISO/IEC 23007-2"
   id="MPEG-U-AUI.xsd">
      <import namespace="urn:mpeg:mpeg7:schema:2004" schemaLocation="mp74U.xsd"/>
```

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

```
</schema>
```

### 5.1.2 Basic data types

This Subclause describes structure of the basic data types which are commonly used in this part of ISO/IEC 23007 as a basic building block of the tools.

#### 5.1.2.1 Syntax

```
<!-- ########################################################### -->
<!-- Vector Datatype                                              -->
<!-- ########################################################### -->
<complexType name="VectorType">
   <sequence>
      <element name="X" type="float"/>
      <element name="Y" type="float"/>
      <element name="Z" type="float" minOccurs="0"/>
   </sequence>
</complexType>
<!-- ########################################################### -->
<!-- Chirality Datatype                                            -->
<!-- ########################################################### -->
<simpleType name="ChiralityType">
   <restriction base="string">
      <enumeration value="Right"/>
      <enumeration value="Left"/>
   </restriction>
</simpleType>
<!-- ########################################################### -->
<!-- AUI Base Datatype                                           -->
<!-- ########################################################### -->
<complexType name="AUIBaseType">
   <attribute name="capturedTimeStamp" type="unsignedLong" default="0"
use="optional"/>
   <attribute name="userId" type="ID" use="optional"/>
</complexType>
<!-- ################################################ -->
<!-- Composite Pattern                               -->
<!-- ################################################ -->
```

```
<element name="CompositePattern" type="aui:CompositePatternType"/>
<complexType name="CompositePatternType">
   <complexContent>
      <extension base="aui:AUIBaseType">
         <sequence>
            <element name="AUIPattern" type="aui:AUIBaseType"
maxOccurs="unbounded"/>
         </sequence>
         <attribute name="sameObject" type="boolean" default="false"
use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

### 5.1.2.2 Semantics

Semantics of the basic datatypes :

| *Name* | *Definition* |
|---|---|
| `VectorType` | This type describes the vector type composed of two float values and one optional value to represent a set of values. |
| `X` | A Value that describes a float value(can be force, torque, position) for x-axis |
| `Y` | A Value that describes a float value(can be force, torque, position) for y-axis |
| `Z` | A Value that describes a float value(can be force, torque, position) for z-axis |
| `ChiralityType` | This type describes whether the hand of interest is a left hand or a right hand.<br>The value of “Right” describes that the hand is a right hand and the value of “Left” describes that the hand is a left hand. |
| `AUIBaseType` | This type provides the topmost type of the base type hierarchy which each individual AUI data formats can inherit. |
| `capturedTimeStamp` | This attribute specifies the time (in milliseconds relative to the epoch) at which a user interaction was captured. When the value of capturedTimeStamp is not available, a value of 0 will be returned. Examples of epoch time are the time of the system start or 0:0:0 UTC 1st January 1970. The `capturedTimeStamp` shall be used as defined in W3C DOM Events. |
| `userId` | This attribute describes an index referencing the user who is generating AUI patterns. |
| `CompositePatternType` | This type provides the container type as the placeholder for the set of AUI Patterns. Since users may generate more than one AUI patterns simultaneously, this type helps to transfer that set of patterns. |

| *Name* | *Definition* |
|---|---|
| `AUIPattern` | This element specifies the AUI patterns that are simultaneously captured and transmitted. All patterns which inherit AUIBaseType can be contained. |
| `sameObject` | This attribute to indicate whether the patterns in a composite pattern are generated from same object or not. |

### 5.1.3 Reference coordinate system

This part of ISO/IEC 23007 extends the coornidate system of W3C DOM Events. All user interactions are projected to a 2D or 3D screen space and the screen provides a unit-based coordinate system to represent where a user interaction happens. For instance, a general 2D monitor screen has (0,0) position at the top-left corner and (number of horizontal units, number of vertical units) position at the bottom-right corner for mice. Also, this part of ISO/IEC 23007 can describe the depth information which is the distance between the origin of the screen and the position where a user interaction happens. The unit of the reference coordinate system can be pixel for current screen coordinate or millimeter for real mesures.

## 5.2 Geometric pattern

### 5.2.1 Introduction

In general, users generate kinds of geometric behaviours in a space with their fingers, hands and even whole body. Advanced physical sensors capture that geometric information as 2D or 3D Cartesian positions. However all position information of sensed feature points can be recognized as one of 2D or 3D geometric objects. Therefore this Subclause provides basic geometric patterns as below:

| *Name* | *Meaning* | *features* |
|---|---|---|
| Point | A geometric point in 2D or 3D Euclidean space | Cartesian 2D or 3D position, (x,y) or (x,y,z). In this part of ISO/IEC 23007, all positions are described as Cartesian 2D or 3D positions. |
| Line | A straight pattern between two points | Two positions at the two ends of a line with optional values of starting timestamp, average velocity and maximum acceleration. |
| Rect | A crossed quadrilateral pattern with four angles | Positions of two opposite corners and two optional corners with four optional timestamps to represent when each corner was drawn. |
| Arc | a curved segment pattern of the circumference of a circle | Two positions for the two end points of the circumference of a circle and the position of the circle center point with optional values of an average angular velocity and a starting timestamp. |
| Circle | A closed curve pattern which is specified with a set of points which have same distance from a center point and which divides the plane into two regions, an interior and an exterior. | Position of the circle center point and the value of the circle's radius with optional values of an average angular velocity and a starting timestamp |

### 5.2.2 Syntax

```
<!-- ################################################ -->
<!-- Point Pattern                         -->
<!-- ################################################ -->
<element name="Point" type="aui:PointType"/>
<complexType name="PointType">
   <complexContent>
      <extension base="aui:AUIBaseType">
         <sequence>
            <element name="Position" type="aui:VectorType"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ################################################ -->
<!-- Line Pattern                                    -->
<!-- ################################################ -->
<element name="Line" type="aui:LineType"/>
<complexType name="LineType">
   <complexContent>
      <extension base="aui:AUIBaseType">
         <sequence>
            <element name="FirstPosition" type="aui:VectorType"/>
            <element name="SecondPosition" type="aui:VectorType"/>
         </sequence>
         <attribute name="startingTimeStamp" type="float" default=”0”
use="optional"/>
         <attribute name="averageVelocity" type="float" default=”0”
use="optional"/>
         <attribute name="maxAcceleration" type="float" default=”0”
use="optional"/>
      </extension>
   </complexContent>
</complexType>
<!-- ################################################ -->
<!-- Rect Pattern                                    -->
<!-- ################################################ -->
<element name="Rect" type="aui:RectType"/>
<complexType name="RectType">
   <complexContent>
      <extension base="aui:AUIBaseType">
         <sequence>
            <element name="TopLeftPosition" type="aui:VectorType"
minOccurs="0"/>
            <element name="BottomRightPosition" type="aui:VectorType"
minOccurs="0"/>
            <element name="TopRightPosition" type="aui:VectorType"
minOccurs="0"/>
            <element name="BottomLeftPosition" type="aui:VectorType"
minOccurs="0"/>
         </sequence>
         <attribute name="firstTimeStamp" default=”0” type="float"
use="optional"/>
         <attribute name="secondTimeStamp" default=”0” type="float"
use="optional"/>
         <attribute name="thirdTimeStamp" default=”0” type="float"
use="optional"/>
         <attribute name="forthTimeStamp" default=”0” type="float"
use="optional"/>
```

```
        </extension>
    </complexContent>
</complexType>
<!-- ############################################### -->
<!-- Arc Pattern                            -->
<!-- ############################################### -->
<element name="Arc" type="aui:ArcType"/>
<complexType name="ArcType">
    <complexContent>
        <extension base="aui:AUIBaseType">
            <sequence>
                <element name="FirstPosition" type="aui:VectorType"/>
                <element name="SecondPosition" type="aui:VectorType"/>
                <element name="CenterPosition" type="aui:VectorType"/>
            </sequence>
            <attribute name="startingTimeStamp" type="float" default=”0”
use="optional"/>
            <attribute name="averageAngularVelocity" default=”0” type="float"
use="optional"/>
        </extension>
    </complexContent>
</complexType>
<!-- ############################################### -->
<!-- Circle Pattern                         -->
<!-- ############################################### -->
<element name="Circle" type="aui:CircleType"/>
<complexType name="CircleType">
    <complexContent>
        <extension base="aui:AUIBaseType">
            <sequence>
                <element name="CenterPosition" type="aui:VectorType"/>
                <element name="Radius" type="float"/>
            </sequence>
            <attribute name="startingTimeStamp" type="float" default=”0”
use="optional"/>
            <attribute name="averageAngularVelocity" type="float" default=”0”
use="optional"/>
        </extension>
    </complexContent>
</complexType>
```

### 5.2.3 Semantics

Semantics of the `Point`, `Line`, `Rect`, `Arc` and `Circle` type:

| *Name* | *Definition* |
|---|---|
| `PointType` | This type describes a geometric point pattern in 2D or 3D Euclidean space. |
| `Position` | This element describes a Cartesian 2D or 3D position using VectorType, (X,Y) or (X,Y,Z) at which the event occurred relative to the origin of the screen coordinate system. ex: A position of a finger, a hand, head or even body. |
| `LineType` | This type describes a line pattern which consists of two end points. |

| *Name* | *Definition* |
|---|---|
| FirstPosition | This element describes a Cartesian 2D or 3D position to represent the position of one end point in a line pattern at which the event occurred relative to the origin of the screen coordinate system. |
| SecondPosition | This element describes a Cartesian 2D or 3D position to represent the position of the other end point in a line pattern at which the event occurred relative to the origin of the screen coordinate system. |
| startingTimeStamp | This attribute describes timing information when drawing a line pattern was started. |
| averageVelocity | This attribute describes the value of average velocity while creating a line pattern. |
| maxAcceleration | This attribute describes the value of maximum acceleration while creating a line pattern. |
| RectType | This type describes a rectangular pattern which consists of four corner positions. A rectangular can be determined with at least two positions of a pair of opposite corners or four positions of rectangle's four corners. |
| TopLeftPosition | This element describes the position at the top-left corner of a rectangular pattern at which the event occurred relative to the origin of the screen coordinate system. |
| BottomRightPosition | This element describes the position at the bottom-right corner of a rectangular pattern at which the event occurred relative to the origin of the screen coordinate system. |
| TopRightPosition | This element describes the position at the top-right corner of a rectangular pattern at which the event occurred relative to the origin of the screen coordinate system. |
| BottomLeftPosition | This element describes the position at the bottom-left corner of a rectangular pattern at which the event occurred relative to the origin of the screen coordinate system. |
| firstTimeStamp | This attribute describes the timing information when drawing a rectangular pattern was started. It means this attribute represents when the first corner position was captured. |
| secondTimeStamp | This attribute describes the timing information when the second corner was constructed and one line pattern was detected. |
| thirdTimeStamp | This attribute describes the timing information when the third corner was constructed and connected two line patterns were detected. |
| fourthTimeStamp | This attribute describes the timing information when the fourth corner was constructed and connected three line patterns were detected. |
| ArcType | This type describes an arc pattern which is a segment of the circumference of a circle. |
| FirstPosition | This element describes the Cartesian 2D or 3D position of one end point in an arc pattern at which the event occurred relative to the origin of the screen |

| *Name* | *Definition* |
|---|---|
| | coordinate system. |
| SecondPosition | This element describes the Cartesian 2D or 3D position of the other end point in an arc pattern at which the event occurred relative to the origin of the screen coordinate system. |
| CenterPosition | This element describes the Cartesian 2D or 3D position of a circle center point in an arc pattern at which the event occurred relative to the origin of the screen coordinate system. |
| startingTimeStamp | This attribute describes timing information when drawing an arc pattern was started. |
| averageAngularVelocity | This attribute describes the value of average angular velocity while creating an arc pattern. |
| CircleType | This type describes a circle pattern. |
| CenterPosition | This element describes the Cartesian 2D or 3D position of a circle center point in a circle pattern at which the event occurred relative to the origin of the screen coordinate system. |
| Radius | This element describes the radius of a circle pattern. |
| startingTimeStamp | This attribute describes timing information when drawing a circle pattern was started. |
| averageAngularVelocity | This attribute describes the value of average angular velocity while creating a circle pattern. |

### 5.2.4 Example

This example shows the description of geometric patterns with the following semantics. The first captured pattern was a rectangular pattern. The rect pattern captured at time stamp “3” provides four 2D positions (0.0, 3.0), (4.0,4.0), (4.0, 0.0) and (0.0, 0.0). This example does not include additional timing information. It means that applications which use this example don’t know when the four corner of this pattern was created.

```
<aui:Rect capturedTimeStamp="3"
xsi:schemaLocation="urn:mpeg:mpegu:schema:aui:2012 MPEG-U-AUI.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aui="urn:mpeg:mpegu:schema:aui:2012">
   <aui:TopLeftPosition>
      <aui:X>0.0</aui:X>
      <aui:Y>3.0</aui:Y>
   </aui:TopLeftPosition>
   <aui:BottomRightPosition>
      <aui:X>4.0</aui:X>
      <aui:Y>0.0</aui:Y>
   </aui:BottomRightPosition>
   <aui:TopRightPosition>
      <aui:X>4.0</aui:X>
      <aui:Y>4.0</aui:Y>
   </aui:TopRightPosition>
   <aui:BottomLeftPosition>
```

```
      <aui:X>0.0</aui:X>
      <aui:Y>0.0</aui:Y>
   </aui:BottomLeftPosition>
</aui:Rect>
```

The second captured pattern was a line pattern. The line pattern captured at time stamp “3” provides two 3D positions (0.0, 0.0, 0.0) and (3.0, 0.0, 0.0). This example also includes additional timing information and it means that this pattern was started at 0.0 and captured at 3. During the drawing, the average value of velocity and the maximum value of acceleration were calculated.

```
<aui:Line startingTimeStamp="0.0" capturedTimeStamp="3" maxAcceleration="1.0"
averageVelocity="1.0" xsi:schemaLocation=" urn:mpeg:mpegu:schema:aui:2012 MPEG-U-
AUI.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aui="urn:mpeg:mpegu:schema:aui:2012">
   <aui:FirstPosition>
      <aui:X>0.0</aui:X>
      <aui:Y>0.0</aui:Y>
      <aui:Z>0.0</aui:Z>
   </aui:FirstPosition>
   <aui:SecondPosition>
      <aui:X>3.0</aui:X>
      <aui:Y>0.0</aui:Y>
      <aui:Z>0.0</aui:Z>
   </aui:SecondPosition>
</aui:Line>
```

## 5.3 Symbolic pattern

### 5.3.1 Introduction

Instead of speaking or writing a word, simple gestures help to communicate with others. For instance, ‘O’ sign and ‘V’ sign, which are well-known hand gestures, are already used in various situations. Therefore, this Subclause provides a container format for symbolic patterns and a classification scheme to enlist well-known symbolic shapes.

### 5.3.2 Syntax

```
<!-- ################################################ -->
<!-- Symbolic Pattern                                 -->
<!-- ################################################ -->
<element name="SymbolicPattern" type="aui:SymbolicPatternType"/>
<complexType name="SymbolicPatternType">
   <complexContent>
      <extension base="aui:AUIBaseType">
         <sequence>
            <element name="Position" type="aui:VectorType"/>
            <element name="Size" type="float"/>
         </sequence>
         <attribute name="symbolType" type="mpeg7:termReferenceType"
use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

### 5.3.3 Semantics

Semantics of the SymbolicPattern type:

| *Name* | *Definition* |
|---|---|
| `SymbolicPatternType` | This type describes a symbolic pattern container.<br>ex. V sign, okay sign, heart sign |
| `Position` | This element describes the Cartesian 2D or 3D position to represent where a symbolic pattern was captured and the event occurred relative to the origin of the screen coordinate system. |
| `Size` | This element describes the size value of a symbolic pattern. |
| `symbolType` | This attribute describes the label of a symbolic pattern as a reference to a classification scheme term provided by SymbolTypeCS. |

### 5.3.4 SymbolTypeCS

This Subclause specifies a classification scheme to indicate the types of a symbolic pattern that may be used by applications using description tools specified in this part of ISO/IEC 23007. Applications need not use this classification scheme; they can use proprietary or third party ones. However, if they choose to use the classification scheme defined in this Subclause, no modifications or extensions are allowed to these classification schemes. The classification scheme in this Subclause is specified using the `ClassificationScheme` defined in ISO/IEC 15938-5. All of the classification schemes defined in this Subclause are uniquely identified by a URN following the "`urn:mpeg:mpeg-u:01-AUI-SymbolType-NS`" namespace identifier.

```
<!-- ################################################ -->
<!-- Symbol Type CS                                   -->
<!-- ################################################ -->
<ClassificationScheme uri="urn:mpeg:mpeg-u:01-AUI-SymbolTypeCS-NS">
   <Term termID="victory">
      <Name xml:lang="en">Victory</Name>
      <Definition xml:lang="en">
Describes the V sign with raised index and middle fingers or raised arms
      </Definition>
   </Term>
   <Term termID="heart">
      <Name xml:lang="en">Heart</Name>
      <Definition xml:lang="en">
Describes the heart shape with hands or arms.
      </Definition>
   </Term>
   <Term termID="rock">
      <Name xml:lang="en">Rock</Name>
      <Definition xml:lang="en">
Describes the rock sign in rock, scissors and paper game.
      </Definition>
   </Term>
   <Term termID="scissors">
      <Name xml:lang="en">Scissors</Name>
      <Definition xml:lang="en">
```

```
Describes the scissors sign in rock, scissors and paper game.
      </Definition>
   </Term>
   <Term termID="paper">
      <Name xml:lang="en">Paper</Name>
      <Definition xml:lang="en">
Describes the paper sign in rock, scissors and paper game.
      </Definition>
   </Term>
   <Term termID="okay">
      <Name xml:lang="en">Okay</Name>
      <Definition xml:lang="en">
Describes the okay sign by connecting the thumb and forefinger in a circle and
holding the other fingers straight or making a circle with two arms
      </Definition>
   </Term>
</ClassificationScheme>
```

### 5.3.5 Example

This example shows the description of a symbolic pattern with the following semantics. This pattern was recognized as one of hand gestures, paper. This pattern captured at time stamp “3” provides a position (3.0, 3.0, -1.0).

```
<aui:SymbolicPattern capturedTimeStamp="3" symbolType="paper"
xsi:schemaLocation="urn:mpeg:mpegu:schema:aui:2012 MPEG-U-AUI.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aui="urn:mpeg:mpegu:schema:aui:2012">
   <aui:Position>
      <aui:X>3.0</aui:X>
      <aui:Y>3.0</aui:Y>
      <aui:Z>-1.0</aui:Z>
   </aui:Position>
   <aui:Size>1.0</aui:Size>
</aui:SymbolicPattern>
```

## 5.4 Touch pattern

### 5.4.1 Introduction

Instead of classical user interfaces such as mouse and keyboard, multi touch interfaces are already applied at various devices. Moreover, many applications adopt well-known touch patterns to control themselves. In this Subclause, a container format for well-known touch patterns and a classification scheme to enlist basic touch patterns are provided.

### 5.4.2 Syntax

```
<!-- ################################################ -->
<!-- Touch pattern                         -->
<!-- ################################################ -->
<element name="TouchPattern" type="aui:TouchPatternType"/>
<complexType name="TouchPatternType">
   <complexContent>
      <extension base="aui:AUIBaseType">
         <sequence>
```

```
                <element name="Position" type="aui:VectorType"/>
            </sequence>
            <attribute name="touchType" type="mpeg7:termReferenceType"
use="optional"/>
            <attribute name="value" type="float" use="optional"/>
        </extension>
    </complexContent>
</complexType>
```

### 5.4.3 Semantics

Semantics of the TouchPattern type:

<table>
<tr><th><i>Name</i></th><th><i>Definition</i></th></tr>
<tr><td><code>TouchPatternType</code></td><td>This type describes a touch pattern container.<br>ex. Tap, Double tap, Flick</td></tr>
<tr><td><code>Position</code></td><td>This element describes the Cartesian 2D or 3D position to represent where a touch pattern was captured and the event occurred relative to the origin of the screen coordinate system.</td></tr>
<tr><td><code>touchType</code></td><td>This attribute describes the label of a symbolic touch pattern as a reference to a classification scheme term provided by TouchTypeCS.</td></tr>
<tr><td><code>value</code></td><td>This attribute describes the value that a touch pattern needs. It means that the meaning of this attribute is dependent on the touch pattern as following
<table>
<tr><th><i>TouchType</i></th><th><i>Meaning of value</i></th></tr>
<tr><td>TapType</td><td>n/a</td></tr>
<tr><td>DoubleTapType</td><td>The value of duration between two taps with respect to millisecond</td></tr>
<tr><td>PressType</td><td>The value of pressed duration in a pressing with respect to millisecond</td></tr>
<tr><td>DragType</td><td>The value of angle between a dragging line and horizontal line</td></tr>
<tr><td>RotateType</td><td>Radian distance between starting position and ending position</td></tr>
<tr><td>FlickType</td><td>n/a</td></tr>
</table>
</td></tr>
</table>

### 5.4.4 TouchTypeCS

This Subclause specifies a classification scheme to indicate the types of a symbolic touch pattern that may be used by applications using description tools specified in this part of ISO/IEC 23007. Applications need not use this classification scheme; they can use proprietary or third party ones. However, if they choose to use the classification scheme defined in this Subclause, no modifications or extensions are allowed to these classification schemes. The classification scheme in this Subclause is specified using the `ClassificationScheme` defined in ISO/IEC 15938-5. All of the classification schemes defined in this Subclause are uniquely identified by a URN following the "`urn:mpeg:mpeg-u:01-AUI-TouchType-NS`" namespace identifier.

```
<!-- ################################################ -->
<!-- Touch Type CS                                   -->
<!-- ################################################ -->
<ClassificationScheme uri="urn:mpeg:mpeg-u:01-AUI-TouchTypeCS-NS">
   <Term termID="tap">
      <Name xml:lang="en">Tap</Name>
      <Definition xml:lang="en">
Describes the briefly touching surface with fingertip.
       </Definition>
   </Term>
   <Term termID="doubletap">
      <Name xml:lang="en">Double Tap</Name>
      <Definition xml:lang="en">
Describes the rapidly touching surface twice with fingertip..
      </Definition>
   </Term>
   <Term termID="press">
      <Name xml:lang="en">Press</Name>
      <Definition xml:lang="en">
Describes the touching surface for extended period of time.
      </Definition>
   </Term>
<Term termID="drag">
      <Name xml:lang="en">Drag</Name>
      <Definition xml:lang="en">
Describes the moving fingertip over surface without losing contact.
      </Definition>
   </Term>
   <Term termID="rotate">
      <Name xml:lang="en">Rotate</Name>
      <Definition xml:lang="en">
Describes the touching surface and moving in a clockwise or counterclockwise
direction.
      </Definition>
   </Term>
   <Term termID="flick">
      <Name xml:lang="en">Flick</Name>
      <Definition xml:lang="en">
Describes the quickly brushing surface with fingertip
       </Definition>
   </Term>
</ClassificationScheme>
```

### 5.4.5 Example

This example shows the description of a touch pattern with the following semantics. This pattern was recognized as one of hand gestures, rotating touch pattern. This pattern was started at a position (1.0, 1.0, -1.0) and captured at time stamp “3” with “-1.0” radian value of a rotating movement.

```
<aui:TouchPattern capturedTimeStamp="3" touchType="rotate" value="-
1.0"  xsi:schemaLocation="urn:mpeg:mpegu:schema:aui:2012 MPEG-U-AUI.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aui="urn:mpeg:mpegu:schema:aui:2012">
    <aui:Position>
        <aui:X>1.0</aui:X>
        <aui:Y>1.0</aui:Y>
        <aui:Z>-1.0</aui:Z>
    </aui:Position>
</aui:TouchPattern>
```

## 5.5 Hand posture pattern

### 5.5.1 Introduction

This Subclause describes the hand posture pattern to support the intuitive hand based interaction for scene description. For example, if a user wants to control an object in a scene description, the user in the real life makes a hand posture such as grab, fist, and opened palm which would be a good candidate posture to support such an interaction modality.

### 5.5.2 Syntax

```
<!-- ################################################ -->
<!--  Hand Posture Pattern                         -->
<!-- ################################################ -->
<element name="HandPosture" type="aui:HandPostureType"/>
<complexType name="HandPostureType">
    <complexContent>
        <extension base="aui:AUIBaseType">
            <sequence>
                <element name="Posture" type="aui:HandPostureBaseType"
maxOccurs="2"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="HandPostureBaseType">
    <sequence>
        <element name="PostureType" type="aui:HandPostureDataType"/>
        <element name="Chirality" type="aui:ChiralityType" minOccurs="0"/>
        <element name="Position" type="aui:VectorType" minOccurs="0"/>
    </sequence>
</complexType>
<!-- ################################################ -->
<!--   HandPostureDataType           -->
<!-- ################################################ -->
<simpleType name="HandPostureDataType">
    <restriction base="mpeg7:termReferenceType"/>
</simpleType>
```

### 5.5.3 Semantics

Semantics of the HandPosture type:

| *Name* | *Definition* |
|---|---|
| `HandPostureType` | This type describes a posture event of user’s hand. |
| `Posture` | This element describes a posture type of user’s hand. |
| `HandPostureBaseType` | This type defines a base type for describing a hand posture. |
| `PostureType` | This element describes a posture of hand from a posture set enumerated in hand posture classification scheme. |
| `Chirality` | This element describes whether the hand of interest is a left hand or a right hand. |
| `Position` | This element describes a position of user’s hand at which the event occurred relative to the origin of the screen coordinate system. |

### 5.5.4 HandPostureTypeCS

This Subclause specifies a classification scheme to indicate the types of hand posture patterns those may be used by applications using description tools specified in this part of ISO/IEC 23007. Applications need not use this classification scheme; they can use proprietary or third party ones. However, if they choose to use the classification scheme defined in this Subclause, no modifications or extensions are allowed to this classification schemes. The classification scheme in this Subclause is specified using the `ClassificationScheme` defined in ISO/IEC 15938-5. All of the classification scheme defined in this Subclause is uniquely identified by a URN following the "`urn:mpeg:mpeg-u:01-AUI-HandPostureType-NS`" namespace identifiers.

```
<!-- ################################################ -->
<!-- Hand Posture Type CS                                   -->
<!-- ################################################ -->
<ClassificationScheme uri="urn:mpeg:mpeg-u:01-AUI-HandPostureCS-NS">
   <Term termID=“open_palm">
      <Name xml:lang="en">Open Palm</Name>
      <Definition xml:lang="en">
         Describes the openness of the palm of the hand
      </Definition>
   </Term>
   <Term termID=“fist">
      <Name xml:lang="en">Fist</Name>
      <Definition xml:lang="en">
         Describes a hand when it is tightly closed with the fingers bent into
the palm.
      </Definition>
   </Term>
   <Term termID=“pointing">
      <Name xml:lang="en">Pointing</Name>
      <Definition xml:lang="en">
         Describes a hand with an index finger stretched out.
      </Definition>
   </Term>
   <Term termID=“thumb_up">
```

```
        <Name xml:lang="en">Thumb-Up</Name>
        <Definition xml:lang="en">
            Describes a hand with a thumb pointing upward and the rest fingers bent
into the palm.
        </Definition>
    </Term>
    <Term termID=“thumb_down">
        <Name xml:lang="en">Thumb-Down</Name>
        <Definition xml:lang="en">
            Describes a hand with a thumb pointing downward and the rest fingers
bent into the palm.
        </Definition>
    </Term>
    <Term termID=“grab">
        <Name xml:lang="en">Grab</Name>
        <Definition xml:lang="en">
            Describes a hand when all the finger tips are in contact.
        </Definition>
    </Term>
    <Term termID=“not_determined">
        <Name xml:lang="en">Not determined</Name>
        <Definition xml:lang="en">
            Describes a hand when the hand posture are not in any specified
postures.
        </Definition>
    </Term>
</ClassificationScheme>
```

### 5.5.5 Example

This example shows the description of hand posture pattern with the following semantics. The first captured pattern was an open palm pattern. The open palm pattern captured at time stamp “3” with the position of (3.0, 3.0, -1.0).

```
<aui:HandPosture capturedTimeStamp="3" Posture="open_palm"
xsi:schemaLocation="urn:mpeg:mpegu:schema:aui:2012 MPEG-U-AUI.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aui="urn:mpeg:mpegu:schema:aui:2012">
    <aui:Position>
        <aui:X>3.0</aui:X>
        <aui:Y>3.0</aui:Y>
        <aui:Z>-1.0</aui:Z>
    </aui:Position>
</aui:HandPosture>
```

## 5.6 Hand gesture pattern

### 5.6.1 Introduction

Due to the fast commercialization of vision based interaction devices, various interaction modalities on consumer electronic devices will prevail over the conventional interaction devices. The real time full body gesture estimation is ready to be used for such a novel interaction. Moreover, the 3D display with/without glasses is now capable of visualizing graphical objects in front of the screen. Such 3D effects provide a desire for users to touch or grab an object in 3D display.

This Subclause describes the hand based interaction methods to meet such needs. The proposed methods contain hand gesture-based interaction. For example, when the user wants to interact with people using hand gestures, “waving a hand” gesture is common semantic to all the human beings, even though the contextual meaning of the gesture may vary.

### 5.6.2 Syntax

```
<!-- ################################################ -->
<!--        Hand Gesture Information                  -->
<!-- ################################################ -->
<element name="HandGesture" type="aui:HandGestureType"/>
<complexType name="HandGestureType">
   <complexContent>
      <extension base="aui:AUIBaseType">
         <sequence>
            <element name="Gesture" type="aui:HandGestureDataType"
minOccurs=”0”/>
            <element name="Chirality" type="aui:ChiralityType" minOccurs="0"/>
         </sequence>
      </extension >
   </complexContent>
</complexType>

<!-- ################################################ -->
<!--   Hand Gesture Data Type             -->
<!-- ################################################ -->
<simpleType name="HandGestureDataType">
   <restriction base="mpeg7:termReferenceType"/>
</simpleType>
```

### 5.6.3 Semantics

Semantics of the HandGestureType:

| *Name* | *Definition* |
|---|---|
| HandGestureType | This type describes a gesture event of user’s hand. |
| Gesture | This element describes the gesture type of user’s hand. |
| Chirality | This element describes whether the hand of interest is a left hand or a right hand. |
| HandGestureDataType | This type describes a gesture of user’s hand from the gesture set enumerated in the classification scheme of HandGestureCS. |

### 5.6.4 HandGestureCS

```
<ClassificationScheme uri="urn:mpeg:mpeg-u:01-AUI-HandGestureCS-NS">
   <Term termID=“push">
      <Name xml:lang="en">Push</Name>
      <Definition xml:lang="en">
         Describes a hand motion when the hand moves forward.
```

```
          </Definition>
   </Term>
   <Term termID=“pull">
       <Name xml:lang="en">Pull</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand moves backward.
          </Definition>
   </Term>
   <Term termID=“slap_left">
       <Name xml:lang="en">Slap Left</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand moves left.
          </Definition>
   </Term>
   <Term termID=“slap_right">
       <Name xml:lang="en">Slap Right</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand moves right.
          </Definition>
   </Term>
   <Term termID=“slap_top">
       <Name xml:lang="en">Slap Top</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand moves upward.
          </Definition>
   </Term>
   <Term termID=“slap_bottom">
       <Name xml:lang="en">Slap Bottom</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand moves downward.
          </Definition>
   </Term>
   <Term termID=“circle_clockwise">
       <Name xml:lang="en">Circle Clockwise</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand draws a clockwise circle.
          </Definition>
   </Term>
   <Term termID=“circle_anti-clockwise">
       <Name xml:lang="en">Circle Anti-clockwise</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand draws a anti-clockwise circle.
          </Definition>
   </Term>
   <Term termID=“waving">
       <Name xml:lang="en">Waving</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand is raised and moved back and
forth.
          </Definition>
   </Term>
   <Term termID=“check">
       <Name xml:lang="en">Check</Name>
       <Definition xml:lang="en">
          Describes a hand motion when the hand draws a check mark.
          </Definition>
   </Term>
</ClassificationScheme>
```

### 5.6.5 Examples

Example 1: This example shows the description of the hand gesture pattern with the following semantics. The first captured pattern was a “slap left” gesture. The “slap left” pattern captured at time stamp “3”.

```
<aui:HandGesture capturedTimeStamp="3" Gesture="slap_left"
xsi:schemaLocation="urn:mpeg:mpegu:schema:aui:2012 MPEG-U-AUI.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aui="urn:mpeg:mpegu:schema:aui:2012">
</aui:HandGesture>
```

Example 2: This example shows the description of a hand gesture pattern parameterized by a line with the following semantics. The captured gesture pattern was a “slap left” gesture of the left hand with the parameterized line which has the first position of (0.0, 0,0, 0,0) and the second position of (3.0, 0.0, 0.0) with the average speed of 10.0 and the maximum acceleration of 1.4.

```
<aui:HandGestureCompositePattern xmlns:aui="urn:mpeg:mpegu:schema:aui:2012"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpegu:schema:aui:2012 MPEG-U-AUI.xsd"
xsi:type="aui:CompositePatternType">
   <aui:AUIPattern xsi:type="aui:HandGestureType" >
      <aui:Gesture>slap-left</aui:Gesture>
      <aui:Chirality>Left</aui:Chirality>
   </aui:AUIPattern>
   <aui:AUIPattern xsi:type="aui:LineType" averageVelocity="10.0"
maxAcceleration="1.4">
      <aui:FirstPosition>
         <aui:X>0.0</aui:X>
         <aui:Y>0.0</aui:Y>
         <aui:Z>0.0</aui:Z>
      </aui:FirstPosition>
      <aui:SecondPosition>
         <aui:X>3.0</aui:X>
         <aui:Y>0.0</aui:Y>
         <aui:Z>0.0</aui:Z>
      </aui:SecondPosition>
   </aui:AUIPattern>
</aui:HandGestureCompositePattern>
```

# Annex A
(informative)

## Relationship between MPEG-U and MPEG-V

Within the MPEG-U Framework, the use of MPEG-V is achieved by means of a mapping of MPEG-V Sensed Information Description Tools and Device Command Description Tools onto an MPEG-U defined interface as conceptually shown in Figure 5. The scopes of and the relationship among related MPEG standards (MPEG-U Part 1&2 and MPEG-V Part 5) are also depicted in Figure 5. The clarification is that MPEG-V Part 5 provides description tools for low-level data coming/going to real world devices. MPEG-U provides either an automatic mapping (using message in/out mapping) to use this data in the scene or description tools that are closer (semantically) to the scene, possibly resulting from a transformation of MPEG-V Part 5 descriptions (semantics generator or UI format interpreter acting as an MPEG-V adaptation engine).

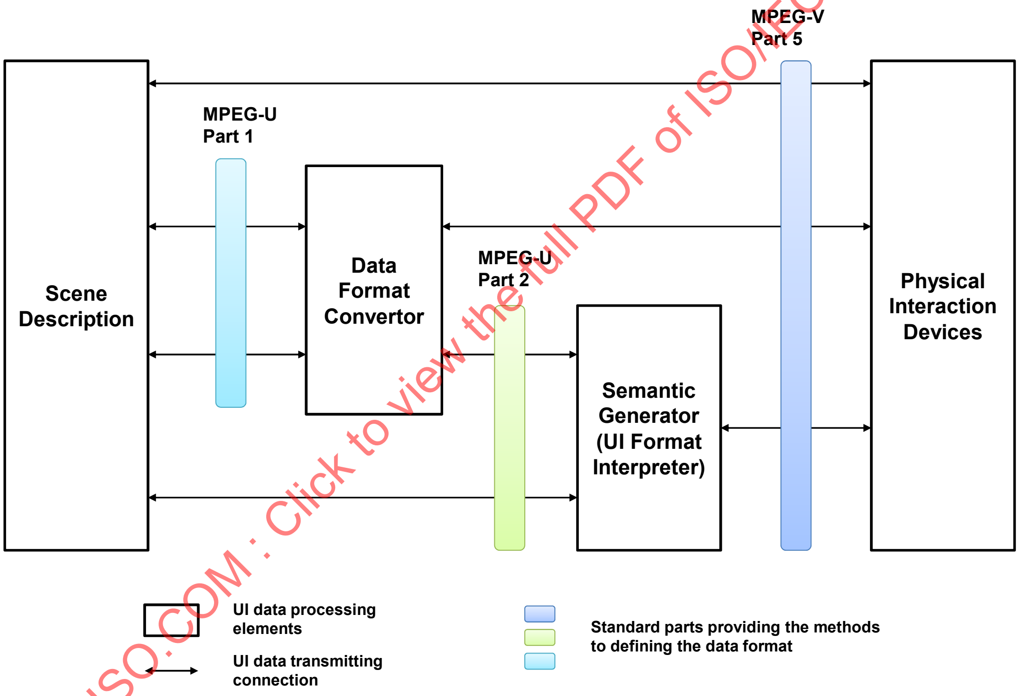

Figure A.1 — High-level view of the relationship between MPEG-U and MPEG-V

This interface should be provided as part 2 of MPEG-U as follows:

The interface for mapping MPEG-V Sensed Information Description Tools and Device Command Description Tools is identified by the type “urn:mpeg:mpegu:schema:widgets:mpegv:2010”. Input messages correspond to messages that are sent by a Sensor to the MPEG-U Widget Manager and output messages are messages sent to the device commands by the Widget Manager. The name of the MPEG-U message is the name of the type of MPEG-V description tool. And for each parameter in the MPEG-V description tools, an input or output parameter is added to the message with the same name.

Example:

```
<messageIn name="LightSensorType">
     <input name="Activate" scriptParamType="number"/>
     <input name="Timestamp" scriptParamType="number"/>
     <input name="Lifespan" scriptParamType="number"/>
     <input name="value" scriptParamType="number"/>
     <input name="Color" scriptParamType="number"/>
</messageIn>

<messageOut name="LightType">
     <output name="activate" scriptParamType="Boolean"/>
     <output name="intensity" scriptParamType="number"/>
     <output name="color" scriptParamType="number"/>
</messageOut>
```

# Annex B
(informative)

# Predefined Message Interfaces of AUI for Widget Manager

## B.1 Introduction

ISO/IEC 23007-1 MPEG-U part 1, specifies a widget management method in terms of widget packaging, communication, and life cycle management. According to Figure 5, data instances of this part can be transported via widget communication method of ISO/IEC 23007-1. Therefore, this annex specifies a set of message interfaces which defines the urn, the predefined interfaces of the AUI patterns. Through these message interfaces, a scene description can receive the events of AUI patterns without knowing anything about AUI devices.

## B.2 URN of the predefined message interfaces

To identify the specific interface for this part, an urn, “urn:mpeg:mpegu:schema:widgets:aui:2011” represents the interface as following:

```
<mw:interface type="urn:mpeg:mpegu:schema:widgets:aui:2011">
<!--
Detail description of interfaces to be transpoted
-->
</mw:interface>
```

## B.3 Message interfaces of AUI patterns

Basically, element and attribute names of input interfaces are same as those of AUI patterns in section 5 while semantics of message interfaces are not different. Though many of input interfaces are optional in the schema description, there is no indicator in the definition of message interfaces in ISO/IEC 23007-1. Therefore widgets and widget managers should deal with those interfaces as optional.

```
<messageIn name="Point">
     <input name="capturedTimeStamp" scriptParamType="number"/>
     <input name="userId" scriptParamType="string"/>
     <input name="x" scriptParamType="number"/>
     <input name="y" scriptParamType="number"/>
     <input name="z" scriptParamType="number"/>
</messageIn>

<messageIn name="Line">
     <input name="capturedTimeStamp" scriptParamType="number"/>
     <input name="userId" scriptParamType="string"/>
     <input name="firstPositionX" scriptParamType="number"/>
     <input name="firstPositionY" scriptParamType="number"/>
     <input name="firstPositionZ" scriptParamType="number"/>
     <input name="secondPositionX" scriptParamType="number"/>
     <input name="secondPositionY" scriptParamType="number"/>
     <input name="secondPositionZ" scriptParamType="number"/>
     <input name="startingTimeStamp" scriptParamType="number"/>
     <input name="averageVelocity" scriptParamType="number" />
     <input name="maxAcceleration" scriptParamType="number" />
</messageIn>
<messageIn name="Rect">
     <input name="capturedTimeStamp" scriptParamType="number"/>
     <input name="userId" scriptParamType="string"/>
     <input name="topLeftPositionX" scriptParamType="number"/>
     <input name="topLeftPositionY" scriptParamType="number"/>
     <input name="topLeftPositionZ" scriptParamType="number"/>
     <input name="bottomRightPositionX" scriptParamType="number"/>
     <input name="bottomRightPositionY" scriptParamType="number"/>
     <input name="bottomRightPositionZ" scriptParamType="number"/>
     <input name="topRightPositionX" scriptParamType="number"/>
     <input name="topRightPositionY" scriptParamType="number"/>
     <input name="topRightPositionZ" scriptParamType="number"/>
     <input name="bottomLeftPositionX" scriptParamType="number"/>
     <input name="bottomLeftPositionY" scriptParamType="number"/>
     <input name="bottomLeftPositionZ" scriptParamType="number"/>
     <input name="firstTimeStamp" scriptParamType="number" />
     <input name="secondTimeStamp" scriptParamType="number" />
     <input name="thirdimeStamp" scriptParamType="number" />
     <input name="fourthTimeStamp" scriptParamType="number" />
</messageIn>

<messageIn name="Arc">
     <input name="capturedTimeStamp" scriptParamType="number"/>
     <input name="userId" scriptParamType="string"/>
     <input name="firstPositionX" scriptParamType="number"/>
     <input name="firstPositionY" scriptParamType="number"/>
     <input name="firstPositionZ" scriptParamType="number"/>
     <input name="secondPositionX" scriptParamType="number"/>
     <input name="secondPositionY" scriptParamType="number"/>
     <input name="secondPositionZ" scriptParamType="number"/>
     <input name="centerPositionX" scriptParamType="number"/>
     <input name="centerPositionY" scriptParamType="number"/>
     <input name="centerPositionZ" scriptParamType="number"/>
     <input name="startingTimeStamp" scriptParamType="number" />
     <input name="averageAngularVelocity" scriptParamType="number" />
</messageIn>

<messageIn name="Circle">
     <input name="capturedTimeStamp" scriptParamType="number"/>
     <input name="userId" scriptParamType="string"/>
     <input name="centerPositionX" scriptParamType="number"/>
     <input name="centerPositionY" scriptParamType="number"/>
     <input name="centerPositionZ" scriptParamType="number"/>
     <input name="radius" scriptParamType="number"/>
     <input name="startingTimeStamp" scriptParamType="number" />
     <input name="averageAngularVelocity" scriptParamType="number" />
</messageIn>

<messageIn name="SymbolicPattern">
     <input name="capturedTimeStamp" scriptParamType="number"/>
     <input name="userId" scriptParamType="string"/>
     <input name="PositionX" scriptParamType="number"/>
     <input name="PositionY" scriptParamType="number"/>
     <input name="PositionZ" scriptParamType="number"/>
     <input name="size" scriptParamType="number"/>
     <input name="symbolType" scriptParamType="string" />
</messageIn>
```