
**Information technology — Media
context and control —**

**Part 7:
Conformance and reference software**

*Technologies de l'information — Contrôle et contexte de supports —
Partie 7: Conformité et logiciel de référence*

IECNORM.COM : Click to view the full PDF of ISO/IEC 23005-7:2017

IECNORM.COM : Click to view the full PDF of ISO/IEC 23005-7:2017



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms, definitions and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Abbreviated terms	2
4 Reference software for the ISO/IEC 23005 series	2
4.1 Overview	2
4.2 ISO/IEC 23005-2 APIs	3
4.2.1 Overview	3
4.2.2 CIM engine	3
4.2.3 CIM creation	3
4.2.4 CIM access	3
4.3 ISO/IEC 23005-3 APIs	3
4.3.1 Overview	3
4.3.2 SEM engine	3
4.3.3 SEM creation	4
4.3.4 SEM access	4
4.4 ISO/IEC 23005-4 APIs	4
4.4.1 Overview	4
4.4.2 VWOC engine	4
4.4.3 VWOC creation	4
4.4.4 VWOC access	4
4.5 ISO/IEC 23005-5 APIs	4
4.5.1 Overview	4
4.5.2 IID engine	5
4.5.3 IID creation	5
4.5.4 IID access	5
4.6 Binary representation APIs for the ISO/IEC 23005 series	5
4.6.1 Overview	5
4.6.2 BinaryIO	5
4.6.3 DefaultBinaryIO	5
5 Conformance for the ISO/IEC 23005 series	6
5.1 General	6
5.2 Rule-based conformance for the ISO/IEC 23005 series	6
5.2.1 Overview	6
5.2.2 Validation schema	6
5.2.3 Description	17
5.2.4 Conformance bit-streams	22
5.3 Schema-based conformance for the ISO/IEC 23005 series	41
5.3.1 Overview	41
5.3.2 Example Valid CIM	41
5.3.3 Example Not Valid CIM	42
5.3.4 Example Valid SEM	42
5.3.5 Example Not Valid SEM	42
5.3.6 Example Valid VWOC	43
5.3.7 Example Not Valid VWOC	43
5.3.8 Example Valid IIM	43
5.3.9 Example Not Valid IIM	43
Bibliography	44

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 World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by ISO/IEC Joint Technical Committee 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 23005-7:2014), which has been technically revised.

A list of all parts in the ISO/IEC 23005 series can be found on the ISO website.

Introduction

This document specifies the conformance and reference software for the ISO/IEC 23005 series. The conformance and reference software serves three main purposes:

- validation of the written specification of the several parts of ISO/IEC 23005;
- clarification of the written specification of the several parts of ISO/IEC 23005;
- conformance testing for checking interoperability for the various applications against the reference software which aims to be compliant with ISO/IEC 23005.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23005-7:2017

IECNORM.COM : Click to view the full PDF of ISO/IEC 23005-7:2017

Information technology — Media context and control —

Part 7:

Conformance and reference software

1 Scope

This document specifies the conformance and reference software implementing the normative clauses of all parts of ISO/IEC 23005. The information provided is applicable for determining the reference software modules available for the parts of ISO/IEC 23005, understanding the functionality of the available reference software modules, and utilizing the available reference software modules. The available reference software modules are specified in the form of application programming interfaces (API) according to ISO/IEC 23006-1.

Furthermore, this document provides means for conformance testing, i.e. bit-streams (XML descriptions) that conform or do not conform to the normative clauses of the other parts of ISO/IEC 23005 and informative descriptions thereof.

2 Normative references

The following documents are referred to in text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

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

ISO/IEC 23005-4, *Information technology — Media context and control — Part 4: Virtual world object characteristics*

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

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23006-1 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.2 Abbreviated terms

API	application programming interface
CIM	control information metadata
MXM	MPEG extensible middleware
SEM	sensory effect metadata
VWOC	virtual world object characteristics
IIM	interaction information metadata

4 Reference software for the ISO/IEC 23005 series

4.1 Overview

This subclause specifies the reference software for the ISO/IEC 23005 series, which is written in Java^[5] and adopts the following package structure:

<i>Name</i>	<i>Definition</i>
org	Java package name for reference software provided by organizations such as ISO/IEC, W3C, or similar.
org.iso	Java package name for reference software provided by ISO/IEC.
org.iso.mpeg	Java package name for reference software provided by ISO/IEC JTC 1/SC 29.
org.iso.mpeg.mpegv	Java package name for reference software provided in the course of the development of the ISO/IEC 23005 series. NOTE 1 Subsequent packages for the individual ISO/IEC 23005 parts use the uncapitalized abbreviations as defined in 3.2, e.g. — org.iso.mpeg.mpegv.cidl, org.iso.mpeg.mpegv.dcdv, org.iso.mpeg.mpegv.sapv, org.iso.mpeg.mpegv.scdv and org.iso.mpeg.mpegv.sepv for ISO/IEC 23005-2, — org.iso.mpeg.mpegv.sedl and org.iso.mpeg.mpegv.sev for ISO/IEC 23005-3, — org.iso.mpeg.mpegv.vwoc for ISO/IEC 23005-4, or — org.iso.mpeg.mpegv.iidl, org.iso.mpeg.mpegv.dcv and org.iso.mpeg.mpegv.siv for ISO/IEC 23005-5. — org.iso.mpeg.mpegv.ct for ISO/IEC 23005-6. NOTE 2 Code included within these packages can be generated automatically by using tools such as JAXB.
org.iso.mpeg.mpegv.{part}.binary	Java package name for binary representation of reference software provided in the course of the development of the ISO/IEC 23005 series. the {part} placeholder is replaced by the component of each part, i.e. cidl, dcdv, sapv, scdv, sepv, sedl, sev, vwoc, iidl, dcv, siv and ct.
org.iso.mpeg.mxm	Java package name for reference software provided in the course of the development of ISO/IEC 23006-1. The actual API is defined within this package.

4.2 ISO/IEC 23005-2 APIs

4.2.1 Overview

This subclause specifies the API to the ISO/IEC 23005-2 reference software. The API is defined in Java and adopts the following package structure:

<i>Name</i>	<i>Definition</i>
<code>org.iso.mpeg.mxm.engine.cimengine</code>	Java package name for API to the ISO/IEC 23005-2 reference software.

4.2.2 CIM engine

The CIMEngine interface defines the methods for operating over data structures as defined within ISO/IEC 23005-2. Classes implementing the CIM engine interface act as factories creating instances of classes performing the following functionalities:

- classes to create data structures, by means of the CIM creation engine;
- classes to access data contained in data structures, by means of CIM parser engine.

4.2.3 CIM creation

Creating CIM structures involves the following interfaces:

- CIMCreator: an interface defining the methods to create CIM structures.

4.2.4 CIM access

Accessing CIM structures involves the following interfaces:

- CIMParser: an interface defining the methods to parse CIM structures.

4.3 ISO/IEC 23005-3 APIs

4.3.1 Overview

This subclause specifies the API to the ISO/IEC 23005-3 reference software. The API is defined in Java and adopts the following package structure:

<i>Name</i>	<i>Definition</i>
<code>org.iso.mpeg.mxm.engine.semengine</code>	Java package name for API to the ISO/IEC 23005-3 reference software.

4.3.2 SEM engine

The SEMEngine interface defines the methods for operating over data structures as defined within ISO/IEC 23005-3. Classes implementing the SEM engine interface act as factories creating instances of classes performing the following functionalities:

- classes to create data structures, by means of the SEM creation engine;
- classes to access data contained in data structures, by means of SEM parser engine.

4.3.3 SEM creation

Creating SEM structures involves the following interfaces:

- SEMCreator: an interface defining the methods to create SEM structures.

4.3.4 SEM access

Accessing SEM structures involves the following interfaces:

- SEMParser: an interface defining the methods to parse SEM structures.

4.4 ISO/IEC 23005-4 APIs

4.4.1 Overview

This subclause specifies the API to the ISO/IEC 23005-4 reference software. The API is defined in Java and adopts the following package structure:

Name	Definition
org.iso.mpeg.mxm.engine.vwocengine	Java package name for API to the ISO/IEC 23005-4 reference software.

4.4.2 VWOC engine

The VWOC Engine interface defines the methods for operating over data structures as defined within ISO/IEC 23005-4. Classes implementing the VWOC engine interface act as factories creating instances of classes performing the following functionalities:

- classes to create data structures, by means of the VWOC creation engine;
- classes to access data contained in data structures, by means of VWOC parser engine.

4.4.3 VWOC creation

Creating VWOC structures involves the following interfaces:

- VWOC Creator: an interface defining the methods to create VWOC structures.

4.4.4 VWOC access

Accessing VWOC structures involves the following interfaces:

- VWOC Parser: an interface defining the methods to parse VWOC structures.

4.5 ISO/IEC 23005-5 APIs

4.5.1 Overview

This subclause specifies the API to the ISO/IEC 23005-5 reference software. The API is defined in Java and adopts the following package structure:

Name	Definition
org.iso.mpeg.mxm.engine.iidengine	Java package name for API to the ISO/IEC 23005-5 reference software.

4.5.2 IID engine

The IIDEngine interface defines the methods for operating over data structures as defined within ISO/IEC 23005-2. Classes implementing the IID engine interface act as factories creating instances of classes performing the following functionalities:

- classes to create data structures, by means of the IID creation engine;
- classes to access data contained in data structures, by means of IID parser engine.

4.5.3 IID creation

Creating IID structures involves the following interfaces:

- IIDCreator: an interface defining the methods to create IID structures.

4.5.4 IID access

Accessing IID structures involves the following interfaces:

- IIDParser: an interface defining the methods to parse IID structures.

4.6 Binary representation APIs for the ISO/IEC 23005 series

4.6.1 Overview

This subclause specifies the API to the ISO/IEC 23005 series reference software for binary representation. The API is defined in Java and adopts the following package structure:

Name	Definition
org.iso.mpeg.mpegv.binary.core	Java package name for API to the ISO/IEC 23005 series reference software for binary representation.

4.6.2 BinaryIO

The BinaryIO interface defines the methods for operating over data structures as defined within the ISO/IEC 23005 series. Classes implementing the BinaryIO interface act as factories creating instances of classes performing the following functionalities:

- classes to generate a binary structure from a MXM object tree, by means of the construct binary structure function;
- classes to generate a binary file from a binary structure, by means of the write binary function;
- classes to generate a binary structure from a binary structure, by means of the read binary function;
- classes to restore a MXM object tree from a binary structure, by means of the create MXM object function.

4.6.3 DefaultBinaryIO

DefaultBinaryIO class defines function “Encode” and “Decode”. Root elements of other parts (e.g. ISO/IEC 23005-2, ISO/IEC 23005-3, ISO/IEC 23005-4) can be extended from this DefaultBinaryIO class. DefaultBinaryIO involves the following functions:

- encode: a function defining the methods to create binary file from a MXM object tree;
- decode: a function defining the methods to restore a MXM object tree from a binary file.

5 Conformance for the ISO/IEC 23005 series

5.1 General

This subclause defines conformance for the ISO/IEC 23005 series. A bit-stream is compliant to the ISO/IEC 23005 series if it conforms to the conformance definitions specified in [5.2](#) and [5.3](#).

5.2 Rule-based conformance for the ISO/IEC 23005 series

5.2.1 Overview

This subclause defines a validation schema based on ISO/IEC 19757-3 to check the conformance bit-streams (XML descriptions) that conform or do not conform to the normative clauses of the other parts of ISO/IEC 23005 and descriptions thereof. In particular, the additional validation rules as specified in the other parts of ISO/IEC 23005 are checked with this validation schema. The actual schema can be found in [5.2.2](#) and a more readable version of the rules and assertion messages is provided in [5.2.3](#). The actual conformance bit-streams (XML descriptions) are defined in [5.2.4](#). The actual conformance bit-streams and corresponding code can be found attached to this document.

For transforming the validation schema according to ISO/IEC 19757-3 and for validating the examples, Saxon^[6] is used. Saxon is not mandatory and other tools can be used for validating the examples.

5.2.2 Validation schema

```
<schema
  xmlns = "http://purl.oclc.org/dsdl/schematron"
  xmlns:sedl = "urn:mpeg:mpeg-v:2010:01-SEDL-NS"
  xmlns:sev = "urn:mpeg:mpeg-v:2010:01-SEV-NS"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xmlns:si = "urn:mpeg:mpeg21:2003:01-DIA-XSI-NS"
  xmlns:cidl = "urn:mpeg:mpeg-v:2010:01-CIDL-NS"
  xmlns:dcdv = "urn:mpeg:mpeg-v:2010:01-DCDV-NS"
  xmlns:scdv = "urn:mpeg:mpeg-v:2010:01-SCDV-NS"
  xmlns:sepv = "urn:mpeg:mpeg-v:2010:01-SEPV-NS"
  xmlns:mpegvct = "urn:mpeg:mpeg-v:2010:01-CT-NS"
  xmlns:mpeg7 = "urn:mpeg:mpeg7:schema:2004"
  xmlns:dia = "urn:mpeg:mpeg21:2003:01-DIA-NS"
  xmlns:iidl = "urn:mpeg:mpeg-v:2010:01-IIDL-NS"
  xmlns:dcv = "urn:mpeg:mpeg-v:2010:01-DCV-NS"
  xmlns:siv = "urn:mpeg:mpeg-v:2010:01-SIV-NS"
  queryBinding = 'xslt' schemaVersion = 'ISO 19757-3'>
<ns prefix = "sedl" uri = "urn:mpeg:mpeg-v:2010:01-SEDL-NS"/>
<ns prefix = "si" uri = "urn:mpeg:mpeg21:2003:01-DIA-XSI-NS"/>
<ns prefix = "sev" uri = "urn:mpeg:mpeg-v:2010:01-SEV-NS"/>
<ns prefix = "xsi" uri = "http://www.w3.org/2001/XMLSchema-instance"/>
<ns prefix = "mpeg7" uri = "urn:mpeg:mpeg7:schema:2004"/>
```

```

<ns prefix = "cidl" uri = "urn:mpeg:mpeg-v:2010:01-CIDL-NS"/>
<ns prefix = "dcdv" uri = "urn:mpeg:mpeg-v:2010:01-DCDV-NS"/>
<ns prefix = "scdv" uri = "urn:mpeg:mpeg-v:2010:01-SCDV-NS"/>
<ns prefix = "sepv" uri = "urn:mpeg:mpeg-v:2010:01-SEPV-NS"/>
<ns prefix = "mpegvct" uri = "urn:mpeg:mpeg-v:2010:01-CT-NS"/>
<ns prefix = "dia" uri = "urn:mpeg:mpeg21:2003:01-DIA-NS"/>
<ns prefix = "iidl" uri = "urn:mpeg:mpeg-v:2010:01-IIDL-NS"/>
<ns prefix = "dcv" uri = "urn:mpeg:mpeg-v:2010:01-DCV-NS"/>
<ns prefix = "siv" uri = "urn:mpeg:mpeg-v:2010:01-SIV-NS"/>
<title>Schema for validating SEDL/CIDL/IIDL descriptions</title>
<!-- ##### -->
<!-- Schema for validating CIDL(Part 2) descriptions -->
<!-- ##### -->
<pattern name = "SensorDeviceCapability element">
  <!-- R1.0: Check the conformance of SensorDeviceCapability -->
  <rule context = "cidl:SensorDeviceCapability">
    <assert test = "if (@minValue and @maxValue) then if (@minValue > @max-
Value) then false() else true() else true()">
      A minValue shall be less than or equal to a maxValue.
    </assert>
  </rule>
</pattern>
<pattern name = "PositionCapability element">
  <!-- R1.1: Check the conformance of PositionCapability -->
  <rule context = "scdv:PositionCapability">
    <assert test = "if (@minValue and @maxValue) then if (@minValue > @max-
Value) then false() else true() else true()">
      A minValue shall be less than or equal to a maxValue.
    </assert>
  </rule>
</pattern>
<pattern name = "OrientationCapability element">
  <!-- R1.2: Check the conformance of OrientationCapability -->
  <rule context = "scdv:OrientationCapability">
    <assert test = "if (@minValue and @maxValue) then if (@minValue > @max-
Value) then false() else true() else true()">

```

```

        A minValue shall be less than or equal to a maxValue.
    </assert>
</rule>
</pattern>
<pattern name = "VelocityCapability element">
    <!-- R1.3: Check the conformance of VelocityCapability -->
    <rule context = "scdv:VelocityCapability">
        <assert test = "if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
            A minValue shall be less than or equal to a maxValue.
        </assert>
    </rule>
</pattern>
<pattern name = "AngularVelocityCapability element">
    <!-- R1.4: Check the conformance of AngularVelocityCapability -->
    <rule context = "scdv:AngularVelocityCapability">
        <assert test = "if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
            A minValue shall be less than or equal to a maxValue.
        </assert>
    </rule>
</pattern>
<pattern name = "AccelerationCapability element">
    <!-- R1.5: Check the conformance of AccelerationCapability -->
    <rule context = "scdv:AccelerationCapability">
        <assert test = "if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
            A minValue shall be less than or equal to a maxValue.
        </assert>
    </rule>
</pattern>
<pattern name = "AngularAccelerationCapability element">
    <!-- R1.6: Check the conformance of AngularAccelerationCapability -->
    <rule context = "scdv:AngularAccelerationCapability">
        <assert test = "if (@minValue and @maxValue) then if (@minValue > @maxValue) then
false() else true() else true()">
            A minValue shall be less than or equal to a maxValue.
        </assert>
    </rule>
</pattern>
<pattern name = "Range element">

```

```

<!-- R2.*: Check the conformance of range -->
<rule context = "scdv:Range">
  <!--R2.0 -->
  <assert test = "if (child::scdv:XminValue > child::scdv:XmaxValue) then false() else
true()" ">
    An XminValue shall be less than or equal to an XmaxValue.
  </assert>    <!-- R2.1 -->
  <assert test = "if (child::scdv:YminValue > child::scdv:YmaxValue) then false() else
true()" ">
    An YminValue shall be less than or equal to an YmaxValue.
  </assert>
  <!-- R2.2 -->
  <assert test = "if (child::scdv:ZminValue > child::scdv:ZmaxValue) then false() else
true()" ">
    A ZminValue shall be less than or equal to a ZmaxValue.
  </assert>
</rule>
</pattern>
<pattern name = "OrientationRange element">
  <!-- R3.*: Check the conformance of OrientationRange -->
  <rule context = "scdv:OrientationRange">
    <!-- R3.0 -->
    <assert test = "if (child::scdv:YawMin > child::scdv:YawMax) then false() else true()" ">
      An YawMin shall be less than or equal to an YawMax.
    </assert>
    <!-- R3.1 -->
    <assert test = "if (child::scdv:PitchMin > child::scdv:PitchMax) then false() else
true()" ">
      A PitchMin shall be less than or equal to a PitchMax.
    </assert>
    <!-- R3.2 -->
    <assert test = "if (child::scdv:RollMin > child::scdv:RollMax) then false() else
true()" ">
      A RollMin shall be less than or equal to a RollMax.
    </assert>
  </rule>
</pattern>

<!-- ##### -->
<!-- Schema for validating SEDL(Part 3) descriptions -->

```

```

<!-- ##### -->
<pattern name = "SEM element">
  <!-- R1.0: Check the conformance of SEM -->
  <rule context = "sedl:SEM">
    <assert test = "@si:timeScale">
      The SEM element shall have a timeScale attribute.
    </assert>
  </rule>
</pattern>
<pattern name = "GroupOfEffects elements">
  <!-- R2.*: Check the conformance of GroupOfEffects -->
  <rule context = "sedl:GroupOfEffects" <!-- R2.0 -->
    <assert test = "if (not(parent::sedl:Declarations) and not(@si:pts or @si:ptsDelta
or @si:absTime)) then false() else true())">
      A GroupOfEffects outside of a Declarations shall have a timestamp (i.e. pts, pts-
Delta, or absTime).
    </assert>
    <!-- R2.1 -->
    <assert test = "if (not(parent::sedl:Declarations) and (@si:pts and @si:absTime))
then false() else true())">
      A GroupOfEffects outside of a Declarations shall not have both a pts and an ab-
sTime at the same time.
    </assert>
    <!-- R2.2 -->
    <assert test = "if (parent::sedl:Declarations and (@si:pts or @si:absTime)) then
false() else true())">
      A GroupOfEffects within a Declarations shall have only a ptsDelta for a times-
tamp.
    </assert>
  </rule>
</pattern>
<pattern name = "Effect elements">
  <!-- R3.*: Check the conformance of Effects -->
  <rule context = "sedl:Effect">
    <!-- R3.0 -->
    <assert test = "@fade or @activate or @duration">
      At least activate, duration, or fade shall be defined.
    </assert>
    <!-- R3.1 -->
    <assert test = "(not(@si:pts or @si:ptsDelta or @si:absTime) and parent::s-
edl:GroupOfEffects) or @si:pts or @si:ptsDelta or @si:absTime">

```

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

</assert>

<!-- R3.2 -->

<assert test = "((@si:pts or @si:ptsDelta or @si:absTime) and not(parent::sedl:GroupOfEffects)) or not(@si:pts or @si:ptsDelta or @si:absTime)">

An effect within a GroupOfEffects shall not have a timestamp (i.e. pts, ptsDelta, or absTime).

</assert>

<!-- R3.3 -->

<assert test = "if (@duration and @activate) then false() else true()">

If duration is defined, activate may not be defined.

</assert>

<!-- R3.4 -->

<assert test = "if (@duration and @fade) then if (not(@activate)) then true() else false() else true()">

If fade and duration are defined, activate may not be defined.

</assert>

<!-- R3.5 -->

<assert test = "if (@fade and not(@intensity-value and @intensity-range)) then false() else true()">

If fade is defined, intensity-value and intensity-range shall be defined.

</assert>

<!-- R3.6 -->

<assert test = "if (some \$retVal in (for \$siblings in following-sibling::sedl:Effect return @xsi:type = \$siblings/@xsi:type and ((@si:pts = \$siblings/@si:pts or @si:ptsDelta = \$siblings/@si:ptsDelta or @si:absTime = \$siblings/@si:absTime) or parent::sedl:GroupOfEffects) and @location = \$siblings/@location) satisfies \$retVal = true()) then false() else true()">

If two (or more) consecutive Effect elements of the same type share the same timestamp (i.e. pts, ptsDelta, or absTime) and location only the latest in their order of appearance shall be used.

</assert>

<!-- R3.7 -->

<assert test = "if ((@intensity-range and not(@intensity-value)) or (@intensity-value and not(@intensity-range))) then false() else true()">

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

</assert>

```

<!-- R3.8 -->
    <assert test = "if (@intensity-value and @intensity-range) then if ((@
intensity-value> = number(tokenize(@intensity-range,'\s+')[5])) and (number(to-
kenize(@intensity-range,'\s+')[6])> = @intensity-value)) then true() else
false() else true()">
        The intensity-value must be within the intensity-range.
    </assert>
<!-- R3.9 -->
    <assert test = "if(child::sev:TrajectorySamples) then if((not(@duration))
and (@timeSamples)) then true() else false() else true()">
        A number of trajectory samples at an update period must have a timeSam-
ples attribute, but duration attribute must not be described.
    </assert>
<!-- R3.10 -->
<assert test = "if(child::sev:PassiveKinestheticForce) then if((not(@duration))
and (@timeSamples)) then true() else false() else true()">
        Passive kinesthetic force effect element must have a timeSamples attrib-
ute in parent element, but duration attribute must not be described.
    </assert>
<!-- R3.11 -->
    <assert test = "if(((@xsi:type = 'sev:PassiveKinestheticForceType') and
(following-sibling::sedl:Effect/@xsi:type['sev:ActiveKinestheticType'])) or ((@
xsi:type = 'sev:ActiveKinestheticType') and (following-sibling::sedl:Effect/@
xsi:type['sev:PassiveKinestheticForceType']))) then false() else true()">
        Passive kinesthetic force effect element must not be described with ac-
tive kinesthetic effect at the same time.
    </assert>
<!-- R3.12 -->
    <assert test = "if (@duration and @fade) then if (@fade > @duration) then
false() else true() else true()">
        If fade and duration are defined, fade must be less or equal to duration.
    </assert>
<!-- R3.13 -->
    <assert test = "if (not(ancestor::sedl:Declarations) and (not(parent::s-
edl:GroupOfEffects) and (@si:pts and @si:absTime))) then false() else true()">
        An Effect shall not have both a pts and an absTime at the same time.
    </assert>
<!-- R3.14 -->
    <assert test = "if (ancestor::sedl:Declarations and (@si:pts or @si:ab-
sTime)) then false() else true()">

```

```

        An Effect within a Declarations shall have only a ptsDelta for a timestamp.
    </assert>
</rule>
</pattern>

<pattern name = "TrajectorySamples element">
    <!-- R4.0 -->
    <rule context = "sev:TrajectorySamples">
<assert test = "if(tokenize(@mpeg7:dim, ' ')[5] = '6') then true() else false()">
    A number of trajectory samples must be in multiples of 6 since 6 samples include three
    positions and three orientations at each period Therefore, number of rows are fixed to 6.
</assert>
    </rule>
</pattern>

<pattern name = "PassiveKinestheticForce element">
    <!-- R5.0 -->
    <rule context = "sev:PassiveKinestheticForce">
<assert test = "if(tokenize(@mpeg7:dim, ' ')[5] = '6') then true() else false()">
    Rows indicate 3 forces and 3 torques. Therefore, the number of row must be '6'.
    </assert>
    </rule>
</pattern>

<pattern name = "Wave element">
    <!-- R6.0: Check the conformance of Wave -->
    <rule context = "sev:Wave">
<assert test = "parent::sedl:Effect/@duration > 0">
    Wave effect element must have duration attribute in parent element.
</assert>
    </rule>
</pattern>

<pattern name = "Shake element">
    <!-- R7.0: Check the conformance of Shake -->
    <rule context = "sev:Shake">
<assert test = "parent::sedl:Effect/@duration > 0">
    Shake effect element must have duration attribute in parent element.
</assert>

```

```

</rule>
</pattern>

<pattern name = "Spin element">
  <!-- R8.0: Check the conformance of Spin -->
  <rule context = "sev:Spin">
<assert test = "parent::sedl:Effect/@duration > 0">
  Spin effect element must have duration attribute in parent element.
</assert>
  </rule>
</pattern>

<pattern name = "MoveToward element">
  <!-- R9.0: Check the conformance of MoveToward -->
  <rule context = "sev:MoveToward">
<assert test = "@directionH or @directionV">
  At least directionH or directionV shall be defined.
</assert>
  </rule>
</pattern>

<pattern name = "Incline element">
  <!-- R10.*: Check the conformance of Incline -->
  <rule context = "sev:Incline">
<!-- R10.0 -->
<assert test = "@pitch or @roll or @yaw">
  At least pitch, roll, or yaw shall be defined.
</assert>
<!-- R10.1 -->
<assert test = "if (@pitch) then if (@pitchSpeed or @pitchAcceleration) then true() else
false() else true()">
  If pitch is defined, pitchSpeed or pitchAcceleration shall be defined.
</assert>
<!-- R10.2 -->
<assert test = "if (@yaw) then if (@yawSpeed or @yawAcceleration) then true() else false() else
true()">
  If yaw is defined, yawSpeed or yawAcceleration shall be defined.
</assert>
<!-- R10.3 -->
<assert test = "if (@roll) then if (@rollSpeed or @rollAcceleration) then true() else false()
else true()">

```

```

    If roll is defined, rollSpeed or rollAcceleration shall be defined.
</assert>
</rule>
</pattern>

<!-- R11.*: Check the conformance of Colour Correction Parameter -->
<pattern name = "Colour correction element">
  <!-- R11.0 -->
  <rule context = "mpegvct:RGB_Value">
    <assert test = "if(count((tokenize(normalize-space(self::*), ' '))) = 3) then
true() else false()">
      The number of RGB_value must be 3 (red, green, blue). Therefore, the
length of vector is fixed to 3.
    </assert>
  </rule>
  <!-- R11.1 -->
  <rule context = "mpegvct:RGB2XYZ_LUT">
    <assert test = "if((tokenize(@mpeg7:dim, ' ')[5] = '3') and (tokenize(@
mpeg7:dim, ' ')[6] = '3')) then true() else false()">
      The size of REG2XYZ_LUT matrix shall be 3 by 3. Therefore, the number of
rows is fixed to 3 and the number of columns is fixed to 3.
    </assert>
  </rule>
  <!-- R11.2 -->
  <rule context = "mpegvct:RGBScalar_Max">
    <assert test = "if(count((tokenize(normalize-space(self::*), ' '))) = 3) then
true() else false()">
      The number of RGBScalar_Max must be 3 (red, green, blue). Therefore, the
length of RGBScalar_Max vector is fixed to 3.
    </assert>
  </rule>
  <!-- R11.3 -->
  <rule context = "mpegvct:Offset_Value">
    <assert test = "if(count((tokenize(normalize-space(self::*), ' '))) = 3) then
true() else false()">
      The number of Offset_Value must be 3 (red offset, green offset, blue off-
set). Therefore, the length of Offset_Value vector is fixed to 3.
    </assert>
  </rule>

```

```

<!-- R11.4 -->
<rule context = "mpegvct:Gain_Offset_Gamma">
  <assert test = "if((tokenize(@mpeg7:dim, ' ')[5] = '3') and (tokenize(@mpeg7:dim, ' ')[6] = '3')) then true() else false()">
    The size of Gain, Offset, and Gamma is 3, respectively. The size of matrix Gain_Offset_Gamma is 3 by 3. Therefore, the number of rows is fixed to 3 and the number of columns is fixed to 3.
  </assert>
</rule>
<!-- R11.5 -->
<rule context = "mpegvct:InverseLUT">
  <assert test = "if((tokenize(@mpeg7:dim, ' ')[5] = '3') and (tokenize(@mpeg7:dim, ' ')[6] = '3')) then true() else false()">
    The size of Inverse_LUT matrix is 3 by 3. Therefore, the number of rows is fixed to 3 and the number of columns is fixed to 3.
  </assert>
</rule>
<!-- R11.6 -->
<rule context = "mpegvct:IDCG_Value">
  <assert test = "if((tokenize(@mpeg7:dim, ' ')[5] = '3') and (tokenize(@mpeg7:dim, ' ')[6] = '2')) then true() else false()">
    The size of IDCG_Value matrix is 3 by 2. Therefore, the number of rows is fixed to 3 and the number of columns is fixed to 2.
  </assert>
</rule>
</pattern>

<!-- R12.*: Check the conformance of ReferenceEffect -->
<pattern name = "ReferenceEffect elements">
  <rule context = "sed1:ReferenceEffect">
    <!-- R12.0 -->
    <assert test = "if (not(ancestor::sed1:Declarations) and not(parent::s-edl:GroupOfEffects) and not(@si:pts or @si:ptsDelta or @si:absTime)) then false() else true()">
      A ReferenceEffect outside of a GroupOfEffects shall have a timestamp (i.e. pts, ptsDelta, or absTime).
    </assert>
    <!-- R12.1 -->
    <assert test = "if (not(ancestor::sed1:Declarations) and (parent::s-edl:GroupOfEffects and (@si:pts or @si:absTime))) then false() else true()">

```

```

    A ReferenceEffect within a GroupOfEffects shall have only a ptsDelta for
    a timestamp.
    </assert>
    <!-- R12.2 -->
    <assert test = "if (not(ancestor::sedl:Declarations) and (not(parent::s-
    edl:GroupOfEffects) and (@si:pts and @si:absTime))) then false() else true()">
        A ReferenceEffect shall not have both a pts and an absTime at the same
        time.
    </assert>
    <!-- R12.3 -->
    <assert test = "if (ancestor::sedl:Declarations and (@si:pts or @si:ab-
    sTime)) then false() else true()">
        A ReferenceEffect within Declarations shall have only a ptsDelta for a
        timestamp.
    </assert>
</rule>
</pattern>

<!-- ##### -->
<!-- Schema for validating IIDL (Part 5) descriptions -->
<!-- ##### -->
<pattern name = "SensedInfo element">
    <!-- R1.*: Check the conformance of SensedInfo -->
    <rule context = "iidl:SensedInfo">
        <!-- R1.0 -->
        <assert test = "if ((@xsi:type = 'siv:MotionSensorType') and
        (count(child::*) = 1)) then false() else true()">
            A MotionSensor shall contain at least one element in addition to a
            timestamp element.
        </assert>
        <!-- R1.1 -->
        <assert test = "if ((@xsi:type = 'siv:IntelligentCameraType') and
        (count(child::*) = 1)) then false() else true()">
            An IntelligentCamera shall contain at least one element in addition to a
            timestamp element.
        </assert>
    </rule>
</pattern>
</schema>

```

5.2.3 Description

5.2.3.1 Description for ISO/IEC 23005-2

Description for SensorDeviceCapability:

No.	Rule	Description
R1.0	if (@minValue and @maxValue) then if (@minValue > @maxValue) then false() else true() else true()	A minValue shall be less than or equal to a maxValue.

Description for PositionCapability:

No.	Rule	Description
R1.1	if (@minValue and @maxValue) then if (@minValue > @maxValue) then false() else true() else true()	A minValue shall be less than or equal to a maxValue.

Description for OrientationCapability:

No.	Rule	Description
R1.2	if (@minValue and @maxValue) then if (@minValue > @maxValue) then false() else true() else true()	A minValue shall be less than or equal to a maxValue.

Description for VelocityCapability:

No.	Rule	Description
R1.3	if (@minValue and @maxValue) then if (@minValue > @maxValue) then false() else true() else true()	A minValue shall be less than or equal to a maxValue.

Description for AngularVelocityCapability:

No.	Rule	Description
R1.4	if (@minValue and @maxValue) then if (@minValue > @maxValue) then false() else true() else true()	A minValue shall be less than or equal to a maxValue.

Description for AccelerationCapability:

No.	Rule	Description
R1.5	if (@minValue and @maxValue) then if (@minValue > @maxValue) then false() else true() else true()	A minValue shall be less than or equal to a maxValue.

Description for AngularAccelerationCapability:

No.	Rule	Description
R1.6	if (@minValue and @maxValue) then if (@minValue > @maxValue) then false() else true() else true()	A minValue shall be less than or equal to a maxValue.

Description for Range:

No.	Rule	Description
R2.0	if (child::scdv:XminValue > child::scdv:XmaxValue) then false() else true()	An XminValue shall be less than or equal to an XmaxValue.
R2.1	if (child::scdv:YminValue > child::scdv:YmaxValue) then false() else true()	An YminValue shall be less than or equal to an YmaxValue.
R2.2	if (child::scdv:ZminValue > child::scdv:ZmaxValue) then false() else true()	A ZminValue shall be less than or equal to a ZmaxValue.

Description for OrientationRange:

No.	Rule	Description
R3.0	if (child::scdv:YawMin > child::scdv:YawMax) then false() else true()	An YawMin shall be less than or equal to an YawMax.
R3.1	if (child::scdv:PitchMin > child::scdv:PitchMax) then false() else true()	A PitchMin shall be less than or equal to a PitchMax.
R3.2	if (child::scdv:RollMin > child::scdv:RollMax) then false() else true()	A RollMin shall be less than or equal to a RollMax.

5.2.3.2 Description for ISO/IEC 23005-3

Description for SEM:

No.	Rule	Description
R1.0	@si:timeScale	The SEM element shall have a timeScale attribute.

Description for GroupOfEffects:

No.	Rule	Description
R2.0	if (not(parent::sedl:Declarations) and not(@si:pts or @si:ptsDelta or @si:absTime)) then false() else true()	A GroupOfEffects outside of a Declarations shall have a timestamp (i.e. pts, ptsDelta, or absTime).
R2.1	if (not(parent::sedl:Declarations) and (@si:pts and @si:absTime)) then false() else true()	A GroupOfEffects outside of a Declarations shall not have both a pts and an absTime at the same time.
R2.2	if (parent::sedl:Declarations and (@si:pts or @si:absTime)) then false() else true()	A GroupOfEffects within a Declarations shall have only a ptsDelta for a timestamp.

Description for Effects:

No.	Rule	Description
R3.0	@fade or @activate or @duration	At least activate, duration, or fade shall be defined.
R3.1	if (not(ancestor::sedl:Declarations) and not(parent::sedl:GroupOfEffects) and not(@si:pts or @si:ptsDelta or @si:absTime)) then false() else true()	An effect outside of a GroupOfEffects shall have a timestamp (i.e. pts, ptsDelta, or absTime).
R3.2	if (not(ancestor::sedl:Declarations) and (parent::sedl:GroupOfEffects and (@si:pts or @si:absTime))) then false() else true()	An Effect within a GroupOfEffects shall have only a ptsDelta for a timestamp.
R3.3	if (@duration and @activate) then false() else true()	If duration is defined, activate may not be defined.
R3.4	if (@duration and @fade) then if (not(@activate)) then true() else false() else true()	If fade and duration are defined, activate may not be defined.
R3.5	if (@fade and not(@intensity-value and @intensity-range)) then false() else true()	If fade is defined, intensity-value and intensity-range shall be defined.

No.	Rule	Description
R3.6	if (some \$retVal in (for \$siblings in following-sibling::sedl:Effect return @xsi:type = \$siblings/@xsi:type and (((@si:pts = \$siblings/@si:pts or @si:ptsDelta = \$siblings/@si:ptsDelta or @si:absTime = \$siblings/@si:absTime) or parent::sedl:GroupOfEffects) and @location = \$siblings/@location) satisfies \$retVal = true()) then false() else true()	If two (or more) consecutive Effect elements of the same type share the same timestamp (i.e. pts, ptsDelta, or absTime) and location, only the latest in their order of appearance shall be used. NOTE The other Effect elements can be removed from the description.
R3.7	if (((@intensity-range and not(@intensity-value)) or (@intensity-value and not(@intensity-range))) then false() else true()	If intensity-value is present, intensity-range must be present and vice versa.
R3.8	if (@intensity-value and @intensity-range) then if (((@intensity-value >= number(tokenize(@intensity-range, 's+')[5])) and (number(tokenize(@intensity-range, 's+')[6]) >= @intensity-value)) then true() else false() else true()	The intensity-value must be within the intensity-range.
R3.9	if(child::sev:TrajectorySamples) then if-((not(@duration)) and (@timeSamples)) then true() else false() else true()	A number of trajectory samples at an update period must have a timeSamples attribute, but duration attribute must not be described.
R3.10	if(child::sev:PassiveKinestheticForce) then if-((not(@duration)) and (@timeSamples)) then true() else false() else true()	Passive kinesthetic force effect element must have a timeSamples attribute in parent element, but duration attribute must not be described.
R3.11	if (((@xsi:type = 'sev:PassiveKinestheticForceType') and (following-sibling::sedl:Effect/@xsi:type['sev:ActiveKinestheticType'])) or ((@xsi:type = 'sev:ActiveKinestheticType') and (following-sibling::sedl:Effect/@xsi:type['sev:PassiveKinestheticForceType']))) then false() else true()	Passive kinesthetic force effect element must not be described with active kinesthetic effect at the same time
R3.12	if (@duration and @fade) then if (@fade > @duration) then false() else true() else true()	If fade and duration are defined, fade must be less or equal to duration.
R3.13	if (not(ancestor::sedl:Declarations) and (not(parent::sedl:GroupOfEffects) and (@si:pts and @si:absTime))) then false() else true()	An Effect shall not have both a pts and an absTime at the same time.
R3.14	if (ancestor::sedl:Declarations and (@si:pts or @si:absTime)) then false() else true()	An Effect within a Declarations shall have only a pts-Delta for a timestamp.

Description for TrajectorySamples:

No.	Rule	Description
R4.0	if(tokenize(@mpeg7:dim, '')[5] = '6') then true() else false()	A number of trajectory samples must be in multiples of 6 since 6 samples include three positions and three orientations at each period. Therefore, number of rows are fixed to 6.

Description for PassiveKinestheticForce:

No.	Rule	Description
R5.0	if(tokenize(@mpeg7:dim, '')[5] = '6') then true() else false()	Rows indicate 3 forces and 3 torques. Therefore, the number of row must be "6".

Description for Wave:

No.	Rule	Description
R6.0	parent::sedl:Effect/@duration > 0	Wave effect element must have duration attribute in parent element.

Description for Shake:

No.	Rule	Description
R7.0	parent::sedl:Effect/@duration > 0	Shake effect element must have duration attribute in parent element.

Description for Spin:

No.	Rule	Description
R8.0	parent::sedl:Effect/@duration > 0	Spin effect element must have duration attribute in parent element.

Description for MoveToward:

No.	Rule	Description
R9.0	@directionH or @directionV	At least directionH or directionV shall be defined.

Description for Incline:

No.	Rule	Description
R10.0	@pitch or @roll or @yaw	At least pitch, roll, or yaw shall be defined.
R10.1	if (@pitch) then if (child::sev:PitchSpeed or child::sev:PitchAcceleration) then true() else false() else true()	If pitch is defined, PitchSpeed or PitchAcceleration shall be defined.
R10.2	if (@yaw) then if (child::sev:YawSpeed or child::sev:YawAcceleration) then true() else false() else true()	If yaw is defined, YawSpeed or YawAcceleration shall be defined.
R10.3	if (@roll) then if (child::sev:RollSpeed or child::sev:RollAcceleration) then true() else false() else true()	If roll is defined, RollSpeed or RollAcceleration shall be defined.

Description for RGB_Value:

No.	Rule	Description
R11.0	if(count((tokenize(normalize-space(self::*), ' '))) = 3) then true() else false()	The number of RGB_value must be 3 (red, green, blue). Therefore, the length of vector is fixed to 3.

Description for RGB2XYZ_LUT:

No.	Rule	Description
R11.1	if((tokenize(@mpeg7:dim, ' ')[5] = '3') and (tokenize(@mpeg7:dim, ' ')[6] = '3')) then true() else false()	The size of REG2XYZ_LUT matrix shall be 3 by 3. Therefore, the number of rows is fixed to 3 and the number of columns is fixed to 3.

Description for RGBScalar_Max:

No.	Rule	Description
R11.2	if(count((tokenize(normalize-space(self::*), ' '))) = 3) then true() else false()	The number of RGBScalar_Max must be 3 (red, green, blue). Therefore, the length of RGBScalar_Max vector is fixed to 3.

Description for Offset_Value:

No.	Rule	Description
R11.3	if(count((tokenize(normalize-space(self::*), ' ')) = 3) then true() else false())	The number of Offset_Value must be 3 (red offset, green offset, blue offset). Therefore, the length of Offset_Value vector is fixed to 3.

Description for Gain_Offset_Gamma:

No.	Rule	Description
R11.4	if((tokenize(@mpeg7:dim, ' ')[5] = '3') and (tokenize(@mpeg7:dim, ' ')[6] = '3')) then true() else false()	The size of Gain, Offset, and Gamma is 3, respectively. The size of matrix Gain_Offset_Gamma is 3 by 3. Therefore, the number of rows is fixed to 3 and the number of columns is fixed to 3.

Description for InverseLUT:

No.	Rule	Description
R11.5	if((tokenize(@mpeg7:dim, ' ')[5] = '3') and (tokenize(@mpeg7:dim, ' ')[6] = '3')) then true() else false()	The size of Inverse_LUT matrix is 3 by 3. Therefore, the number of rows is fixed to 3 and the number of columns is fixed to 3.

Description for IDCG_Value:

No.	Rule	Description
R11.6	if((tokenize(@mpeg7:dim, ' ')[5] = '3') and (tokenize(@mpeg7:dim, ' ')[6] = '2')) then true() else false()	The size of IDCG_Value matrix is 3 by 2. Therefore, the number of rows is fixed to 3 and the number of columns is fixed to 2.

Description for ReferenceEffect:

No.	Rule	Description
R12.0	if (not(ancestor::sedl:Declarations) and not(parent::sedl:GroupOfEffects) and not(@si:pts or @si:ptsDelta or @si:absTime)) then false() else true()	A ReferenceEffect outside of a GroupOfEffects shall have a timestamp (i.e. pts, ptsDelta, or absTime).
R12.1	if (not(ancestor::sedl:Declarations) and (parent::sedl:GroupOfEffects and (@si:pts or @si:absTime))) then false() else true()	A ReferenceEffect within a GroupOfEffects shall have only a ptsDelta for a timestamp.
R12.2	if (not(ancestor::sedl:Declarations) and (not(parent::sedl:GroupOfEffects) and (@si:pts and @si:absTime))) then false() else true()	A ReferenceEffect shall not have both a pts and an absTime at the same time.
R12.3	if (ancestor::sedl:Declarations and (@si:pts or @si:absTime)) then false() else true()	A ReferenceEffect within Declarations shall have only a ptsDelta for a timestamp.

5.2.3.3 Description for ISO/IEC 23005-5

Description for SensedInfo:

No.	Rule	Description
R1.0	if (((@xsi:type = 'siv:MotionSensorType') and (count(child::*) = 1)) then false() else true())	A MotionSensor shall contain at least one element in addition to a timestamp element.
R1.1	if (((@xsi:type = 'siv:IntelligentCameraType') and (count(child::*) = 1)) then false() else true())	An IntelligentCamera shall contain at least one element in addition to a timestamp element.

5.2.3.4 Validation schema transformation

For transforming the validation schema, the following commands are used.

NOTE The validation schema is called schematron.xsd.

```
java -jar saxon/saxon9.jar -versionmsg:off -s:schematron.xsd
-o:tmp/new_schema1.sch -xsl:schematron/iso_dsdl_include.xsl
java -jar saxon/saxon9.jar -versionmsg:off -s:tmp/new_schema1.sch
-o:tmp/new_schema2.sch -xsl:schematron/iso_abstract_expand.xsl
java -jar saxon/saxon9.jar -versionmsg:off -s:tmp/new_schema2.sch
-o:output/val_schema.xsl -xsl:schematron/iso_svrl_for_xslt1.xsl
```

The first two commands resolve inclusions and abstractions. The last command generates the XSLT style sheet which can then be applied to the XML description. The command generates an XSLT 1.0 description.

5.2.4 Conformance bit-streams

The actual conformance bit-streams are organized as follows and referred to as examples.

- The conformance bit-streams — XML description.
- The command to validate the example (both Schematron and conformance software).
- The output generated during the validation.
- Description of erroneous output (relevant parts).

5.2.4.1 Conformance bit-streams for ISO/IEC 23005-2

5.2.4.1.1 Example 1

ex01.xml shows a valid CIM description with the result in result_ex01.xml. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/cidl/ex01.xml
-o:schematron/output/cidl/result_ex01.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/cidl/ex01.xml
```

NOTE It is necessary to provide the input with the absolute path.

5.2.4.1.2 Example 2

ex02.xml shows an invalid CIM description with the result in result_ex02.xml. The errors occur in the SensorDeviceCapability, PositionCapability, OrientationCapability,

VelocityCapability, AngularVelocityCapability, AccelerationCapability, and AngularAccelerationCapability. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/cidl/ex02.xml
-o:schematron/output/cidl/result_ex02.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/cidl/ex02.xml
```

NOTE It is necessary to provide the input with the absolute path.

The errors during validation are shown in the lines 40 to 46, lines 47 to 53, lines 56 to 62, lines 64 to 70, lines 72 to 78, lines 80 to 86, lines 88 to 94, and lines 96 to 102. Note that all errors are the same but for different test rules. The lines 40 and 41, lines 47 and 48, lines 56 and 57, lines 64 and 65, lines 72 and 73, lines 80 and 81, lines 88 and 89, and lines 96 and 97 defined failed assertions. There is also the test which failed depicted: "if (@minValue and @maxValue) then if (@minValue > @maxValue) then false() else true() else true()". The error message: "A minValue shall be less than or equal to a maxValue." is defined in the lines: 44, 51, 60, 68, 76, 84, 92, and 100.

5.2.4.1.3 Example 3

ex03.xml shows a valid CIM description with the result in result_ex03.xml. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/cidl/ex03.xml
-o:schematron/output/cidl/result_ex03.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/cidl/ex03.xml
```

NOTE It is necessary to provide the input with the absolute path.

5.2.4.1.4 Example 4

ex04.xml shows an invalid CIM description with the result in result_ex04.xml. The errors occur in the range. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/cidl/ex04.xml
-o:schematron/output/cidl/result_ex04.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/cidl/ex04.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 51 to 56. Lines 51 and 52 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:XminValue > child::scdv:XmaxValue) then false() else true()". Line 54 defines the error message: "An XminValue shall be less than or equal to an XmaxValue.". Another error during

the validation is shown in the lines 57 to 62. Lines 57 and 58 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:YminValue > child::scdv:YmaxValue) then false() else true()". Line 60 defines the error message: "An YminValue shall be less than or equal to an YmaxValue.". Another error during the validation is shown in the lines 63 to 68. Lines 63 and 64 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:ZminValue > child::scdv:ZmaxValue) then false() else true()". Line 66 defines the error message: "A ZminValue shall be less than or equal to a ZmaxValue.". The next error during the validation is shown in the lines 69 to 75. Lines 69 and 70 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:XminValue > child::scdv:XmaxValue) then false() else true()". Line 73 defines the error message: "An XminValue shall be less than or equal to an XmaxValue." Another error during the validation is shown in the lines 76 to 81. Lines 76 and 77 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:YminValue > child::scdv:YmaxValue) then false() else true()". Line 79 defines the error message: "An YminValue shall be less than or equal to an YmaxValue." Another error during the validation is shown in the lines 82 to 87. Lines 82 and 83 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:ZminValue > child::scdv:ZmaxValue) then false() else true()". Line 85 defines the error message: "A ZminValue shall be less than or equal to a ZmaxValue."

5.2.4.1.5 Example 5

ex05.xml shows a valid CIM description with the result in result_ex05.xml. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/cidl/ex05.xml
-o:schematron/output/cidl/result_ex05.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/cidl/ex05.xml
```

NOTE It is necessary to provide the input with the absolute path.

5.2.4.1.6 Example 6

ex06.xml shows an invalid CIM description with the result in result_ex06.xml. The errors occur in the OrientationRange. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/cidl/ex06.xml
-o:schematron/output/cidl/result_ex06.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/cidl/ex06.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 51 to 56. Lines 51 and 52 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:YawMin> child::scdv:YawMax) then false() else true()". Line 54 defines the error message: "An YawMin shall be less than or equal to an YawMax." Another error during the validation is shown in the lines 58 to 63. Lines 58 and 59 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:PitchMin> child::scdv:PitchMax) then false() else true()". Line 61 defines the error message: "A PitchMin shall be less than or equal to a

PitchMax." Another error during the validation is shown in the lines 64 to 69. Lines 64 and 65 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:RollMin> child::scdv:RollMax) then false() else true()". Line 67 defines the error message: "A RollMin shall be less than or equal to a RollMax." The next error during the validation is shown in the lines 70 to 76. Lines 70 and 71 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:YawMin> child::scdv:YawMax) then false() else true()". Line 74 defines the error message: "An YawMin shall be less than or equal to an YawMax." Another error during the validation is shown in the lines 77 to 82. Lines 77 and 78 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:PitchMin> child::scdv:PitchMax) then false() else true()". Line 80 defines the error message: "A PitchMin shall be less than or equal to a PitchMax." Another error during the validation is shown in the lines 83 to 88. Lines 83 and 84 defined a failed assertion. There is also the test which failed depicted: "if (child::scdv:RollMin> child::scdv:RollMax) then false() else true()". Line 86 defines the error message: "A RollMin shall be less than or equal to a RollMax."

5.2.4.1.7 Example 7

ex07.xml shows a valid CIM description with the result in result_ex07.xml. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/cidl/ex07.xml
-o:schematron/output/cidl/result_ex07.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/cidl/ex07.xml
```

NOTE It is necessary to provide the input with the absolute path.

5.2.4.2 Conformance bit-streams for ISO/IEC 23005-3

5.2.4.2.1 Example 1

ex01.xml shows a valid SEM description with the result in result_ex01.xml. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex01.xml
-o:schematron/output/sedl/result_ex01.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex01.xml
```

NOTE It is necessary to provide the input with the absolute path.

5.2.4.2.2 Example 2

ex02.xml shows an invalid SEM description with the result in result_ex02.xml. The error occurs in the GroupOfEffects. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex02.xml
-o:schematron/output/sedl/result_ex02.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex02.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 51 to 57. Lines 51 and 52 defined a failed assertion. There is also the test which failed depicted: "if (not(parent::sedl:Declarations) and not(@si:pts or @si:ptsDelta or @si:absTime)) then false() else true()". Line 55 defines the error message: "A GroupOfEffects outside of a Declarations shall have a timestamp (i.e., pts, ptsDelta, or absTime)".

5.2.4.2.3 Example 3

ex03.xml shows an invalid SEM description with the result in result_ex03.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex03.xml
-o:schematron/output/sedl/result_ex03.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex03.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 53 to 59. Lines 53 and 54 defined a failed assertion. There is also the test which failed depicted: "if (not(ancestor::sedl:Declarations) and (parent::sedl:GroupOfEffects and (@si:pts or @si:absTime))) then false() else true()". Line 57 defines the error message: "An Effect within a GroupOfEffects shall have only a ptsDelta for a timestamp."

5.2.4.2.4 Example 4

ex04.xml shows an invalid SEM description with the result in result_ex04.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex04.xml
-o:schematron/output/sedl/result_ex04.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex04.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 52 to 58. Lines 52 and 53 defined a failed assertion. There is also the test which failed depicted: "@fade or @activate or @duration". Line 57 defines the error message: "At least activate, duration, or fade shall be defined."

5.2.4.2.5 Example 5

ex05.xml shows an invalid SEM description with the result in result_ex05.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex05.xml
-o:schematron/output/sedl/result_ex05.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex05.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 52 to 58. Lines 52 and 53 defined a failed assertion. There is also the test which failed depicted: "if (not(ancestor::sedl:Declarations) and not(parent::sedl:GroupOfEffects) and not(@si:pts or @si:ptsDelta or @si:absTime)) then false() else true()". Line 56 defines the error message: "An Effect outside of a GroupOfEffects shall have a timestamp (i.e., pts, ptsDelta, or absTime)."

5.2.4.2.6 Example 6

ex06.xml shows an invalid SEM description with the result in result_ex06.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex06.xml
-o:schematron/output/sedl/result_ex06.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex06.xml
```

NOTE It is necessary to provide the input with the absolute path.

The two errors during the validation are shown in the lines 52 to 58 and lines 59 to 64. Lines 52 and 53 defined the first failed assertion. There is also the test which failed depicted: "if (@duration and @activate) then false() else true()". Line 56 defines the error message: "If duration is defined, activate may not be defined." The second assert is defined in the lines 59 and 60. The test was: "if (@duration and @fade) then if (not(@activate)) then true() else false() else true()". Line 64 defines the error message: "If fade and duration are defined, activate may not be defined."

5.2.4.2.7 Example 7

ex07.xml shows an invalid SEM description with the result in result_ex07.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex07.xml
-o:schematron/output/sedl/result_ex07.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex07.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 52 to 58. Lines 52 and 53 defined a failed assertion. There is also the test which failed depicted: "if (@fade and not(@intensity-value and @intensity-range)) then false() else true()". Line 56 defines the error message: "If fade is defined, intensity-value and intensity-range shall be defined."

5.2.4.2.8 Example 8

ex08.xml shows an invalid SEM description with the result in result_ex08.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex08.xml
-o:schematron/output/sedl/result_ex08.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex08.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 52 to 58. Lines 52 and 53 defined a failed assertion. There is also the test which failed depicted: "if (some \$retVal in (for \$siblings in following-sibling::sedl:Effect return @xsi:type = \$siblings/@xsi:type and ((@si:pts = \$siblings/@si:pts or @si:ptsDelta = \$siblings/@si:ptsDelta or @si:absTime = \$siblings/@si:absTime) or parent::sedl:GroupOfEffects) and @location = \$siblings/@location) satisfies \$retVal = true()) then false() else true()". Line 56 defines the error message: "If two (or more) consecutive Effect elements of the same type share the same timestamp (i.e., pts, ptsDelta, or absTime) and location only the latest in their order of appearance shall be used."

5.2.4.2.9 Example 9

ex09.xml shows an invalid SEM description with the result in result_ex09.xml. The error occurs in the SEM. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex09.xml
-o:schematron/output/sedl/result_ex09.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex09.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 49 to 55. Lines 49 and 50 defined a failed assertion. There is also the test which failed depicted: "@si:timeScale". Line 53 defines the error message: "The SEM element shall have a timeScale attribute."

5.2.4.2.10 Example 10

ex10.xml shows a valid SEM description with the result in result_ex10.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex10.xml
-o:schematron/output/sedl/result_ex10.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex10.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 56 to 62. Lines 56 and 57 defined a failed assertion. There is also the test which failed depicted: "@fade or @activate or @duration". Line 60 defines the error message: "At least activate, duration, or fade shall be defined."

5.2.4.2.11 Example 11

ex11.xml shows an invalid SEM description with the result in result_ex11.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex11.xml
-o:schematron/output/sedl/result_ex11.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex11.xml
```

NOTE It is necessary to provide the input with the absolute path.

The two errors during the validation are shown in the lines 52 to 58 and lines 59 to 65. Lines 52 and 53 defined the first failed assertion. There is also the test which failed depicted: "if ((@intensity-range and not(@intensity-value)) or (@intensity-value and not(@intensity-range))) then false() else true()". Line 56 defines the error message: "If intensity-value is present, intensity-range must be present and vice versa." The second assert is defined in the lines 59 and 60. The test was: "if ((@intensity-range and not(@intensity-value)) or (@intensity-value and not(@intensity-range))) then false() else true()". Line 63 defines the error message: "If intensity-value is present, intensity-range must be present and vice versa."

5.2.4.2.12 Example 12

ex12.xml shows an invalid SEM description with the result in result_ex12.xml. The error occurs in the Effect. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex12.xml
-o:schematron/output/sedl/result_ex12.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex12.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 52 to 58. Lines 52 and 53 defined a failed assertion. There is also the test which failed depicted: "if (@intensity-value and @intensity-range) then if ((@intensity-value >= number(tokenize(@intensity-range, '\s+') [5])) and (number(tokenize(@intensity-range, '\s+') [6]) >= @intensity-value)) then true() else false() else true()". Line 56 defines the error message: "The intensity-value must be within the intensity-range."

5.2.4.2.13 Example 13

ex13.xml shows an invalid SEM description with the result in result_ex13.xml. The error occurs in the TrajectorySamples. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex13.xml
-o:schematron/output/sedl/result_ex13.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex13.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 54 to 60. Lines 54 and 55 defined a failed assertion. There is also the test which failed depicted: "if (tokenize(@mpeg7:dim, ' ')[5] = '6') then true() else false()". Line 58 defines the error message: "A number of trajectory samples must be in multiples of 6 since 6 samples include three positions and three orientations at each period Therefore, number of rows are fixed to 6."

5.2.4.2.14 Example 14

ex14.xml shows an invalid SEM description with the result in result_ex14.xml. The error occurs in the TrajectorySamples. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex14.xml
-o:schematron/output/sedl/result_ex14.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex14.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 53 to 58. Lines 53 and 54 defined the failed assertion. The test was: "if(child::sev:TrajectorySamples) then if((not(@duration)) and (@timeSamples)) then true() else false() else true()". Line 56 defines the error message: "A number of trajectory samples at an update period must have a timeSamples attribute, but duration attribute must not be described."

5.2.4.2.15 Example 15

ex15.xml shows an invalid SEM description with the result in result_ex15.xml. The error occurs in the TrajectorySamples. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex15.xml
-o:schematron/output/sedl/result_ex15.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex15.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 52 to 58. Lines 52 and 53 defined a failed assertion. There is also the test which failed depicted: "if(child::sev:TrajectorySamples) then if((not(@duration)) and (@timeSamples)) then true() else false() else true()". Line 56 defines the error message: "A number of trajectory samples at an update period must have a timeSamples attribute, but duration attribute must not be described."

5.2.4.2.16 Example 16

ex16.xml shows an invalid SEM description with the result in result_ex16.xml. The error occurs in the PassiveKinestheticForce. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex16.xml
-o:schematron/output/sedl/result_ex16.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex16.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 55 to 61. Lines 55 and 56 defined a failed assertion. There is also the test which failed depicted: "if (tokenize(@mpeg7:dim, ' ')[5] = '6') then true() else false())". Line 59 defines the error message: "Rows indicate 3 forces and 3 torques. Therefore, the number of row must be '6'."

5.2.4.2.17 Example 17

ex17.xml shows an invalid SEM description with the result in result_ex17.xml. The error occurs in the PassiveKinestheticForce. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex17.xml
-o:schematron/output/sedl/result_ex17.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex17.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 52 to 58. Lines 52 and 53 defined a failed assertion. There is also the test which failed depicted: "if(child::sev:PassiveKinestheticForce) then if((not(@duration)) and (@timeSamples)) then true() else false() else true()". Line 56 defines the error message: "Passive kinesthetic force effect element must have a timeSamples attribute in parent element, but duration attribute must not be described."

5.2.4.2.18 Example 18

ex18.xml shows an invalid SEM description with the result in result_ex18.xml. The error occurs in the passivekinestheticforce and activekinesthetic. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex18.xml
-o:schematron/output/sedl/result_ex18.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex18.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 52 to 58. Lines 52 and 53 defined a failed assertion. There is also the test which failed depicted: "if (((@xsi:type = 'sev:PassiveKinestheticForceType') and (following-sibling::sedl:Effect/@xsi:type['sev:ActiveKinestheticType'])) or ((@xsi:type = 'sev:ActiveKinestheticType') and (following-sibling::sedl:Effect/@xsi:type['sev:PassiveKinestheticForceType']))) then false() else true()". Line 58 defines the error message: "Passive kinesthetic force effect element must not be described with active kinesthetic effect at the same time."

5.2.4.2.19 Example 19

ex19.xml shows a valid SEM description with the result in result_ex19.xml. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex19.xml
-o:schematron/output/sedl/result_ex19.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex19.xml
```

NOTE It is necessary to provide the input with the absolute path.

5.2.4.2.20 Example 20

ex20.xml shows an invalid SEM description with the result in result_ex20.xml. The error occurs in the MoveToward. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex20.xml
-o:schematron/output/sedl/result_ex20.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex20.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 59 to 65. Lines 59 and 60 defined a failed assertion. There is also the test which failed depicted: "@directionH or @directionV". Line 63 defines the error message: "At least directionH or directionV shall be defined."

5.2.4.2.21 Example 21

ex21.xml shows an invalid SEM description with the result in result_ex21.xml. The error occurs in the Incline. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex21.xml
-o:schematron/output/sedl/result_ex21.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex21.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 60 to 66. Lines 60 and 61 defined a failed assertion. There is also the test which failed depicted: "@pitch or @roll or @yaw". Line 64 defines the error message: "At least pitch, roll, or yaw shall be defined."

5.2.4.2.22 Example 22

ex22.xml shows an invalid SEM description with the result in result_ex22.xml. The error occurs in the Incline. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex22.xml
-o:schematron/output/sedl/result_ex22.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex22.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 60 to 66. Lines 60 and 61 defined a failed assertion. There is also the test which failed depicted: "if (@pitch) then if (@pitchSpeed or @pitchAcceleration) then true() else false() else true()". Line 64 defines the

error message: "If pitch is defined, pitchSpeed or pitchAcceleration shall be defined."

5.2.4.2.23 Example 23

ex23.xml shows an invalid SEM description with the result in result_ex23.xml. The error occurs in the Incline. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex23.xml
-o:schematron/output/sedl/result_ex23.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex23.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 60 to 66. Lines 60 and 61 defined a failed assertion. There is also the test which failed depicted: "if (@yaw) then if (@yawSpeed or @yawAcceleration) then true() else false() else true()". Line 64 defines the error message: "If yaw is defined, yawSpeed or yawAcceleration shall be defined."

5.2.4.2.24 Example 24

ex24.xml shows an invalid SEM description with the result in result_ex24.xml. The error occurs in the Incline. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex24.xml
-o:schematron/output/sedl/result_ex24.xml
-xsl:schematron/output/val_schema.xsl
```

Command for the conformance software:

```
ant run -Dvalid = RULE -Dinput = schematron/examples/sedl/ex24.xml
```

NOTE It is necessary to provide the input with the absolute path.

The error during the validation is shown in the lines 60 to 66. Lines 60 and 61 defined a failed assertion. There is also the test which failed depicted: "if (@roll) then if (@rollSpeed or @rollAcceleration) then true() else false() else true()". Line 64 defines the error message: "If yaw is defined, rollSpeed or rollAcceleration shall be defined."

5.2.4.2.25 Example 25

ex25.xml shows an invalid SEM description with the result in result_ex25.xml. The error occurs in the Wave. The command used for that is as follows:

```
java -jar saxon9.jar -versionmsg:off -s:schematron/examples/sedl/ex25.xml
-o:schematron/output/sedl/result_ex25.xml
-xsl:schematron/output/val_schema.xsl
```