

CLUE Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 2, 2015

R. Presta
S P. Romano
University of Napoli
September 29, 2014

**An XML Schema for the CLUE data model
draft-ietf-clue-data-model-schema-07**

Abstract

This document provides an XML schema file for the definition of CLUE data model types.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 2, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	4
2.	Terminology	4
3.	XML Schema	7
4.	<mediaCaptures>	17
5.	<encodingGroups>	18
6.	<captureScenes>	18
7.	<simultaneousSets>	18
8.	<globalViews>	18
9.	<captureEncodings>	18
10.	<mediaCapture>	18
10.1.	captureID attribute	19
10.2.	mediaType attribute	20
10.3.	<captureSceneIDREF>	20
10.4.	<encGroupIDREF>	20
10.5.	<spatialInformation>	20
10.5.1.	<capturePoint>	21
10.5.2.	<captureArea>	22
10.6.	<nonSpatiallyDefinable>	23
10.7.	<content>	23
10.8.	<synchronizationID>	24
10.9.	<policy>	24
10.10.	<maxCaptures>	24
10.11.	<individual>	25
10.12.	<description>	25
10.13.	<priority>	26
10.14.	<lang>	26
10.15.	<mobility>	26
10.16.	<maxCaptureEncodings>	26
10.17.	<relatedTo>	26
10.18.	<view>	26
10.19.	<presentation>	27
10.20.	<capturedPeople>	27
10.20.1.	<personIDREF>	27
11.	Audio captures	27
11.1.	<sensitivityPattern>	28
12.	Video captures	28
12.1.	<embeddedText>	29
13.	Text captures	29
14.	Other capture types	30
15.	<captureScene>	30
15.1.	<sceneInformation>	31
15.2.	<sceneViews>	31
15.3.	sceneID attribute	32
15.4.	scale attribute	32
16.	<sceneView>	32
16.1.	<mediaCaptureIDs>	33
16.2.	sceneViewID attribute	33
17.	<encodingGroup>	33

17.1.	<maxGroupBandwidth>	34
17.2.	<encodingIDList>	34
17.3.	encodingGroupID attribute	34
18.	<simultaneousSet>	35
18.1.	setID attribute	35
18.2.	mediaType attribute	35
18.3.	<mediaCaptureIDREF>	36
18.4.	<sceneViewIDREF>	36
18.5.	<captureSceneIDREF>	36
19.	<globalView>	36
20.	<people>	36
20.1.	<person>	37
20.1.1.	personID attribute	38
20.1.2.	<personInfo>	38
20.1.3.	<personType>	38
21.	<captureEncoding>	38
21.1.	<captureID>	39
21.2.	<encodingID>	39
21.3.	<configuredContent>	39
22.	<clueInfo>	39
23.	XML Schema extensibility	40
23.1.	Example of extension	40
24.	Security considerations	42
25.	IANA considerations	43
25.1.	XML Schema registration	43
25.2.	XML namespace registration	43
26.	Sample XML file	43
27.	MCC example	50
28.	Diff with draft-ietf-clue-data-model-schema-06 version	56
29.	Diff with draft-ietf-clue-data-model-schema-04 version	57
30.	Diff with draft-ietf-clue-data-model-schema-03 version	58
31.	Diff with draft-ietf-clue-data-model-schema-02 version	58
32.	Informative References	58

1. Introduction

This document provides an XML schema file for the definition of CLUE data model types.

The schema is based on information contained in [\[I-D.ietf-clue-framework\]](#). It encodes information and constraints defined in the aforementioned document in order to provide a formal representation of the concepts therein presented. The schema definition is intended to be modified according to changes applied to the above mentioned CLUE document.

The document aims at the definition of a coherent structure for all the information associated with the description of a telepresence scenario. Such information is used within the CLUE protocol messages ([\[I-D.ietf-clue-protocol\]](#)) enabling the dialogue between a Media Provider and a Media Consumer. CLUE protocol messages, indeed, are XML messages allowing (i) a Media Provider to advertise its telepresence capabilities in terms of media captures, capture scenes, and other features envisioned in the CLUE framework, according to the format herein defined and (ii) a Media Consumer to request the desired telepresence options in the form of capture encodings, represented as described in this document.

2. Terminology

This document refers to the same terminology used in [\[I-D.ietf-clue-framework\]](#), except for the "CLUE Participant" definition (which is still under discussion). We briefly recall herein some of the main terms exploited in the document.

Audio Capture: Media Capture for audio. Denoted as AC_n in the example cases in this document.

Camera-Left and Right: For Media Captures, Camera-Left and Camera-Right are from the point of view of a person observing the rendered media. They are the opposite of Stage-Left and Stage-Right.

Capture: Same as Media Capture.

Capture Device: A device that converts audio and video input into an electrical signal, in most cases to be fed into a media encoder.

Capture Encoding: A specific encoding of a Media Capture, to be sent by a Media Provider to a Media Consumer via RTP.

Capture Scene: An abstraction grouping semantically-coupled Media Captures available at the Media Provider's side, representing a precise portion of the local scene that can be transmitted remotely. Capture Scene MAY correspond to a part of the telepresence room or MAY focus only on the presentation media. A Capture Scene is characterized by a set of attributes and by a set of Capture Scene Views.

Capture Scene View: A list of Media Captures of the same media type that constitute a possible representation of a Capture Scene. Media Capture belonging to the same Capture Scene View can be sent simultaneously by the Media Provider.

CLUE Participant: This term is not imported from the framework terminology and should be considered temporary since it is under review. We introduced it for the sake of simplicity in order to identify a generic entity (either an Endpoint or a MCU) making use of the CLUE protocol.

Consumer: Same as Media Consumer.

Encoding or Individual Encoding: The representation of an encoding technology. In the CLUE datamodel, for each encoding it is provided a set of parameters representing the encoding constraints, like for example the maximum bandwidth of the Media Provider the encoding can consume. s

Encoding Group: The representation of a group of encodings. For each group, it is provided a set of parameters representing the constraints to be applied to the group as a whole. An example is the maximum bandwidth that can be consumed when using the contained encodings together simultaneously.

Endpoint The logical point of final termination through receiving, decoding and rendering, and/or initiation through capturing, encoding, and sending of media streams. An endpoint consists of one or more physical devices which source and sink media streams, and exactly one SIP Conferencing Framework Participant (which, in turn, includes exactly one SIP User Agent). Endpoints can be anything from multiscreen/multicamera room controllers to handheld devices.

MCU: Multipoint Control Unit (MCU) - a device that connects two or more endpoints together into one single multimedia conference. An MCU may include a Mixer.

Media: Any data that, after suitable encoding, can be conveyed over RTP, including audio, video or timed text.

Media Capture: A "Media Capture", or simply "Capture", is a source of Media of a single type (i.e., audio or video or text).

Media Stream: The term "Media Stream", or simply "Stream", is used as a synonymous of Capture Encoding.

Media Provider: A CLUE participant (i.e., an Endpoint or a MCU) able to send Media Streams.

Media Consumer: A CLUE participant (i.e., an Endpoint or a MCU) able to receive Media Streams.

Scene: Same as Capture Scene.

Scene View: Same as Capture Scene View.

Stream: Same of Media Stream.

Multiple Content Capture: A Capture that can contain different Media Captures of the same media type. It is denoted as MCC in this document. In the Stream resulting from the MCC, the Stream coming from the encoding of the composing Media Captures can appear simultaneously, if the MCC is the result of a mixing operation, or can appear alternatively over the time, according to a certain switching policy.

Plane of Interest: The spatial plane containing the most relevant subject matter.

Provider: Same as Media Provider.

Render:

Simultaneous Transmission Set: a set of Media Captures of the same media type that can be transmitted simultaneously from a Media Provider.

Single Media Capture: A Capture representing the Media coming from a single-source Capture Device.

Spatial Information: Data about the spatial position of a Capture Device that generate a Single Media Capture within the context of a Capture Scene representing a physical portion of a Telepresence Room.

Stream Characteristics: The union of the features used to describe a Stream in the CLUE environment and in the SIP-SDP environment.

Video Capture: A Media Capture for video.

3. XML Schema

This section contains the CLUE data model schema definition.

The element and attribute definitions are formal representation of the concepts needed to describe the capabilities of a Media Provider and the streams that are requested by a Media Consumer given the Media Provider's ADVERTISEMENT ([[I-D.ietf-clue-protocol](#)]).

The main groups of information are:

<mediaCaptures>: the list of media captures available ([Section 4](#))

<encodingGroups>: the list of encodings groups ([Section 5](#))

<captureScenes>: the list of capture scenes ([Section 6](#))

<simultaneousSets>: the list of simultaneous transmission sets ([Section 7](#))

<globalViews>: the list of global views sets ([Section 8](#))

<people>: meta data about the participants represented in the telepresence session ([Section 20](#)).

<captureEncodings>: the list of instantiated capture encodings ([Section 9](#))

All of the above refers to concepts that have been introduced in [[I-D.ietf-clue-framework](#)] and further detailed in the following of this document.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:clue-info"
  xmlns:tns="urn:ietf:params:xml:ns:clue-info"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:clue-info"
  xmlns:xcard="urn:ietf:params:xml:ns:vcard-4.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="0.7">
```



```
<!-- Import xcard XML schema -->
<xs:import namespace="urn:ietf:params:xml:ns:vcard-4.0"
schemaLocation="xcard.xsd"/>

<!-- ELEMENT DEFINITIONS -->
<xs:element name="mediaCaptures" type="mediaCapturesType"/>
<xs:element name="encodingGroups" type="encodingGroupsType"/>
<xs:element name="captureScenes" type="captureScenesType"/>
<xs:element name="simultaneousSets" type="simultaneousSetsType"/>
<xs:element name="globalViews" type="globalViewsType"/>
<xs:element name="people" type="peopleType"/>

<xs:element name="captureEncodings" type="captureEncodingsType"/>

<!-- MEDIA CAPTURES TYPE -->
<!-- envelope of media captures -->
<xs:complexType name="mediaCapturesType">
  <xs:sequence>
    <xs:element name="mediaCapture" type="mediaCaptureType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- DESCRIPTION element -->
<xs:element name="description">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="lang" type="xs:language"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<!-- MEDIA CAPTURE TYPE -->
<xs:complexType name="mediaCaptureType" abstract="true">
  <xs:sequence>
    <!-- mandatory fields -->
    <xs:element name="captureSceneIDREF" type="xs:IDREF"/>
    <xs:choice>
      <xs:sequence>
        <xs:element name="spatialInformation"
          type="tns:spatialInformationType"/>
      </xs:sequence>
      <xs:element name="nonSpatiallyDefinable" type="xs:boolean" fixed="true"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```



```
</xs:choice>
<!-- for handling multi-content captures: -->
<xs:choice>
  <xs:sequence>
    <xs:element name="synchronizationID" type="xs:ID" minOccurs="0"/>
    <xs:element name="content" type="contentType" minOccurs="0"/>
    <xs:element name="policy" type="xs:string" minOccurs="0"/>
    <xs:element name="maxCaptures" type="maxCapturesType" minOccurs="0"/>
  </xs:sequence>
  <xs:element name="individual" type="xs:boolean" fixed="true"/>
</xs:choice>
<!-- optional fields -->
<xs:element name="encGroupIDREF" type="xs:IDREF" minOccurs="0"/>
<xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="priority" type="xs:unsignedInt" minOccurs="0"/>
<xs:element name="lang" type="xs:language" minOccurs="0"/>
<xs:element name="mobility" type="mobilityType" minOccurs="0"/>
<xs:element name="presentation" type="presentationType" minOccurs="0"/>
<xs:element name="view" type="viewType" minOccurs="0"/>
<xs:element name="capturedPeople" type="capturedPeopleType" minOccurs="0"/>
<xs:element name="maxCaptureEncodings" type="xs:unsignedInt"
minOccurs="0"/>
<xs:element name="relatedTo" type="xs:IDREF" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="captureID" type="xs:ID" use="required"/>
<xs:attribute name="mediaType" type="xs:string" use="required"/>

</xs:complexType>

<!-- CONTENT TYPE -->
<xs:complexType name="contentType">
  <xs:sequence>
    <xs:element name="captureIDREF" type="xs:IDREF"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="sceneViewIDREF" type="xs:IDREF"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<!-- MAX CAPTURES TYPE -->
<xs:complexType name="maxCapturesType">
  <xs:simpleContent>
    <xs:extension base="xs:unsignedInt">
      <xs:attribute name="exactNumber" type="xs:boolean"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```



```
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- CAPTURED PEOPLE TYPE -->
<xs:complexType name="capturedPeopleType">
  <xs:sequence>
    <xs:element name="personIDREF" type="xs:IDREF" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- PEOPLE TYPE -->
<xs:complexType name="peopleType">
  <xs:sequence>
    <xs:element name="person" type="personType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- PERSON TYPE -->
<xs:complexType name="personType">
  <xs:sequence>
    <xs:element name="personInfo" type="xcard:vcardType" maxOccurs="1"
      minOccurs="0"/>
    <xs:element name="personType" type="personTypeType"
      minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="personID" type="xs:ID" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<!-- PERSON TYPE TYPE -->
<xs:simpleType name="personTypeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="chairman"/>
    <xs:enumeration value="vice-chairman"/>
    <xs:enumeration value="minute taker"/>
    <xs:enumeration value="presenter"/>
    <xs:enumeration value="translator"/>
    <xs:enumeration value="timekeeper"/>
    <xs:enumeration value="attendee"/>
  </xs:restriction>
</xs:simpleType>
```



```
<!-- VIEW TYPE -->
<xs:simpleType name="viewType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="room"/>
    <xs:enumeration value="table"/>
    <xs:enumeration value="lectern"/>
    <xs:enumeration value="individual"/>
    <xs:enumeration value="audience"/>
  </xs:restriction>
</xs:simpleType>

<!-- PRESENTATION TYPE -->
<xs:simpleType name="presentationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="slides"/>
    <xs:enumeration value="image"/>
    <xs:enumeration value=""/>
  </xs:restriction>
</xs:simpleType>

<!-- SPATIAL INFORMATION TYPE -->
<xs:complexType name="spatialInformationType">
  <xs:sequence>
    <xs:element name="capturePoint" type="capturePointType" minOccurs="0"/>
    <xs:element name="captureArea" type="captureAreaType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<!-- MOBILITY TYPE -->
<xs:simpleType name="mobilityType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="static"/>
    <xs:enumeration value="dynamic"/>
    <xs:enumeration value="highly-dynamic"/>
  </xs:restriction>
</xs:simpleType>

<!-- TEXT CAPTURE TYPE -->
<xs:complexType name="textCaptureType">
  <xs:complexContent>
    <xs:extension base="tns:mediaCaptureType">
      <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```



```
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<!-- OTHER CAPTURE TYPE -->
<xs:complexType name="otherCaptureType">
  <xs:complexContent>
    <xs:extension base="tns:mediaCaptureType">
      <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<!-- AUDIO CAPTURE TYPE -->
<xs:complexType name="audioCaptureType">
  <xs:complexContent>
    <xs:extension base="tns:mediaCaptureType">
      <xs:sequence>
        <xs:element name="sensitivPattern" type="sensitivPatternType"
          minOccurs="0"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<!-- SENSITIVITY PATTERN TYPE -->
<xs:simpleType name="sensitivPatternType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="uni"/>
    <xs:enumeration value="shotgun"/>
    <xs:enumeration value="omni"/>
    <xs:enumeration value="figure8"/>
    <xs:enumeration value="cardioid"/>
    <xs:enumeration value="hyper-cardioid"/>
  </xs:restriction>
</xs:simpleType>
```

```
<!-- VIDEO CAPTURE TYPE -->
```



```
<xs:complexType name="videoCaptureType">
  <xs:complexContent>
    <xs:extension base="tns:mediaCaptureType">
      <xs:sequence>
        <xs:element ref="embeddedText" minOccurs="0"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- EMBEDDED TEXT ELEMENT -->
<xs:element name="embeddedText">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:boolean">
        <xs:attribute name="lang" type="xs:language"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<!-- CAPTURE SCENES TYPE -->
<!-- envelope of capture scenes -->
<xs:complexType name="captureScenesType">
  <xs:sequence>
    <xs:element name="captureScene" type="captureSceneType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- CAPTURE SCENE TYPE -->
<xs:complexType name="captureSceneType">
  <xs:sequence>
    <xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="sceneInformation" type="xcard:vcardType" minOccurs="0"/>
    <xs:element name="sceneViews" type="sceneViewsType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="sceneID" type="xs:ID" use="required"/>
  <xs:attribute name="scale" type="scaleType" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<!-- SCALE TYPE -->
```



```
<xs:simpleType name="scaleType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="mm"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="noscale"/>
  </xs:restriction>
</xs:simpleType>

<!-- CAPTURE AREA TYPE -->
<xs:complexType name="captureAreaType">
  <xs:sequence>
    <xs:element name="bottomLeft" type="pointType"/>
    <xs:element name="bottomRight" type="pointType"/>
    <xs:element name="topLeft" type="pointType"/>
    <xs:element name="topRight" type="pointType"/>
  </xs:sequence>
</xs:complexType>

<!-- POINT TYPE -->
<xs:complexType name="pointType">
  <xs:sequence>
    <xs:element name="x" type="xs:decimal"/>
    <xs:element name="y" type="xs:decimal"/>
    <xs:element name="z" type="xs:decimal"/>
  </xs:sequence>
</xs:complexType>

<!-- CAPTURE POINT TYPE -->
<xs:complexType name="capturePointType">
  <xs:complexContent>
    <xs:extension base="pointType">
      <xs:sequence>
        <xs:element name="lineOfCapturePoint" type="tns:pointType"
          minOccurs="0"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- SCENE VIEWS TYPE -->
<!-- envelope of scene views of a capture scene -->
<xs:complexType name="sceneViewsType">
  <xs:sequence>
    <xs:element name="sceneView" type="sceneViewType"
      maxOccurs="unbounded"/>
  </xs:sequence>
```



```
</xs:complexType>
```

```
<!-- SCENE VIEW TYPE -->
```

```
<xs:complexType name="sceneViewType">
```

```
  <xs:sequence>
```

```
    <xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
```

```
    <xs:element name="mediaCaptureIDs" type="captureIDListType"/>
```

```
  </xs:sequence>
```

```
  <xs:attribute name="sceneViewID" type="xs:ID" use="required"/>
```

```
</xs:complexType>
```

```
<!-- CAPTURE ID LIST TYPE -->
```

```
<xs:complexType name="captureIDListType">
```

```
  <xs:sequence>
```

```
    <xs:element name="captureIDREF" type="xs:IDREF"
      maxOccurs="unbounded"/>
```

```
  </xs:sequence>
```

```
</xs:complexType>
```

```
<!-- ENCODING GROUPS TYPE -->
```

```
<xs:complexType name="encodingGroupsType">
```

```
  <xs:sequence>
```

```
    <xs:element name="encodingGroup" type="tns:encodingGroupType"
      maxOccurs="unbounded"/>
```

```
  </xs:sequence>
```

```
</xs:complexType>
```

```
<!-- ENCODING GROUP TYPE -->
```

```
<xs:complexType name="encodingGroupType">
```

```
  <xs:sequence>
```

```
    <xs:element name="maxGroupBandwidth" type="xs:unsignedLong"/>
```

```
    <xs:element name="encodingIDList" type="encodingIDListType"/>
```

```
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
```

```
  </xs:sequence>
```

```
  <xs:attribute name="encodingGroupID" type="xs:ID" use="required"/>
```

```
  <xs:anyAttribute namespace="##any" processContents="lax"/>
```

```
</xs:complexType>
```

```
<!-- ENCODING ID LIST TYPE -->
```

```
<xs:complexType name="encodingIDListType">
```

```
  <xs:sequence>
```

```
    <xs:element name="encID" type="xs:string" maxOccurs="unbounded"/>
```

```
  </xs:sequence>
```

```
</xs:complexType>
```

```
<!-- SIMULTANEOUS SETS TYPE -->
```



```
<xs:complexType name="simultaneousSetsType">
  <xs:sequence>
    <xs:element name="simultaneousSet" type="simultaneousSetType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- SIMULTANEOUS SET TYPE -->
<xs:complexType name="simultaneousSetType">
  <xs:sequence>
    <xs:element name="mediaCaptureIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="sceneViewIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="captureSceneIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="setID" type="xs:ID" use="required"/>
  <xs:attribute name="mediaType" type="xs:string"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>

<!-- GLOBAL VIEWS TYPE -->
<xs:complexType name="globalViewsType">
  <xs:sequence>
    <xs:element name="globalView" type="globalViewType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- GLOBAL VIEW TYPE -->
<xs:complexType name="globalViewType">
  <xs:sequence>
    <xs:element name="sceneViewIDREF" type="xs:IDREF"
      maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="globalViewID" type="xs:ID"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>

<!-- CAPTURE ENCODINGS TYPE -->
<xs:complexType name="captureEncodingsType">
  <xs:sequence>
    <xs:element name="captureEncoding" type="captureEncodingType"
```



```
    maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- CAPTURE ENCODING TYPE -->
<xs:complexType name="captureEncodingType">
  <xs:sequence>
    <xs:element name="captureID" type="xs:string"/>
    <xs:element name="encodingID" type="xs:string"/>
    <xs:element name="configuredContent" type="contentType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>

<!-- CLUE INFO ELEMENT -->
<!-- the <clueInfo> envelope can be seen
      as the ancestor of an <advertisement> envelope -->
<xs:element name="clueInfo" type="clueInfoType"/>

<!-- CLUE INFO TYPE -->
<xs:complexType name="clueInfoType">
  <xs:sequence>
    <xs:element ref="mediaCaptures"/>
    <xs:element ref="encodingGroups"/>
    <xs:element ref="captureScenes"/>
    <xs:element ref="simultaneousSets" minOccurs="0"/>
    <xs:element ref="globalViews" minOccurs="0"/>
    <xs:element ref="people" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="clueInfoID" type="xs:ID" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
</xs:schema>
```

Following sections describe the XML schema in more detail.

4. <mediaCaptures>

<mediaCaptures> represents the list of one or more media captures available on the Media Provider's side. Each media capture is represented by a <mediaCapture> element ([Section 10](#)).

5. <encodingGroups>

<encodingGroups> represents the list of the encoding groups organized on the Media Provider's side. Each encoding group is represented by a <encodingGroup> element ([Section 17](#)).

6. <captureScenes>

<captureScenes> represents the list of the capture scenes organized on the Media Provider's side. Each capture scene is represented by a <captureScene> element. ([Section 15](#)).

7. <simultaneousSets>

<simultaneousSets> contains the simultaneous sets indicated by the Media Provider. Each simultaneous set is represented by a <simultaneousSet> element. ([Section 18](#)).

8. <globalViews>

<globalViews> contains a set of alternative representations of all the scenes that are offered by a Media Provider to a Media Consumer. Each alternative is named "global view" and it is represented by a <globalView> element. ([Section 19](#)).

9. <captureEncodings>

<captureEncodings> is a list of capture encodings. It can represent the list of the desired capture encodings indicated by the Media Consumer or the list of instantiated captures on the provider's side. Each capture encoding is represented by a <captureEncoding> element. ([Section 21](#)).

10. <mediaCapture>

According to the CLUE framework, a media capture is the fundamental representation of a media flow that is available on the provider's side. Media captures are characterized (i) by a set of features that are independent from the specific type of medium, and (ii) by a set of features that are media-specific. The features that are common to all media types appear within the media capture type, that has been designed as an abstract complex type. Media-specific captures, such as video captures, audio captures and others, are specialization of that abstract media capture type, as in a typical generalization-specialization hierarchy.

The following is the XML Schema definition of the media capture type:


```
<!-- MEDIA CAPTURE TYPE -->
<xs:complexType name="mediaCaptureType" abstract="true">
  <xs:sequence>
    <!-- mandatory fields -->
    <xs:element name="captureSceneIDREF" type="xs:IDREF"/>
    <xs:choice>
      <xs:sequence>
        <xs:element name="spatialInformation"
          type="tns:spatialInformationType"/>
      </xs:sequence>
      <xs:element name="nonSpatiallyDefinable" type="xs:boolean" fixed="true"/>
    </xs:choice>
    <!-- for handling multi-content captures: -->
    <xs:choice>
      <xs:sequence>
        <xs:element name="synchronizationID" type="xs:ID" minOccurs="0"/>
        <xs:element name="content" type="contentType" minOccurs="0"/>
        <xs:element name="policy" type="xs:string" minOccurs="0"/>
        <xs:element name="maxCaptures" type="maxCapturesType" minOccurs="0"/>
      </xs:sequence>
      <xs:element name="individual" type="xs:boolean" fixed="true"/>
    </xs:choice>
    <!-- optional fields -->
    <xs:element name="encGroupIDREF" type="xs:IDREF" minOccurs="0"/>
    <xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="priority" type="xs:unsignedInt" minOccurs="0"/>
    <xs:element name="lang" type="xs:language" minOccurs="0"/>
    <xs:element name="mobility" type="mobilityType" minOccurs="0"/>
    <xs:element name="presentation" type="presentationType" minOccurs="0"/>
    <xs:element name="view" type="viewType" minOccurs="0"/>
    <xs:element name="capturedPeople" type="capturedPeopleType" minOccurs="0"/>
    <xs:element name="maxCaptureEncodings" type="xs:unsignedInt"
      minOccurs="0"/>
    <xs:element name="relatedTo" type="xs:IDREF" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="captureID" type="xs:ID" use="required"/>
  <xs:attribute name="mediaType" type="xs:string" use="required"/>
</xs:complexType>
```

10.1. captureID attribute

The "captureID" attribute is a mandatory field containing the identifier of the media capture.

10.2. `mediaType` attribute

The "mediaType" attribute is a mandatory attribute specifying the media type of the capture ("audio", "video", "text",...).

10.3. `<captureSceneIDREF>`

`<captureSceneIDREF>` is a mandatory field containing the identifier of the capture scene the media capture is defined in. Indeed, each media capture must be defined within one and only one capture scene. When a media capture is spatially definable, some spatial information is provided along with it in the form of point coordinates (see [Section 10.5](#)). Such coordinates refers to the space of coordinates defined for the capture scene containing the capture.

10.4. `<encGroupIDREF>`

`<encGroupIDREF>` is an optional field containing the identifier of the encoding group the media capture is associated with. Media captures that are not associated with any encoding group can not be instantiated as media streams.

10.5. `<spatialInformation>`

Media captures are divided into two categories: (i) non spatially definable captures and (ii) spatially definable captures.

Captures are spatially definable when at least (i) it is possible to provide the coordinates of the device position within the telepresence room of origin (capture point) together with its capturing direction specified by a second point (point on line of capture), or (ii) it is possible to provide the represented area within the telepresence room, by listing the coordinates of the four co-planar points identifying the plane of interest (area of capture). The coordinates of the abovementioned points must be expressed according to the coordinate space of the capture scene the media captures belongs to.

Non spatially definable captures cannot be characterized within the physical space of the telepresence room of origin. Capture of this kind are for example those related to recordings, text captures, DVDs, registered presentation, or external streams that are played in the telepresence room and transmitted to remote sites.

Spatially definable captures represent a part of the telepresence room. The captured part of the telepresence room is described by means of the `<spatialInformation>` element. By comparing the `<spatialInformation>` element of different media captures within the

same capture scene, a consumer can better determine the spatial relationships between them and render them correctly. Non spatially definable captures do not show in their XML description such element: they are instead characterized by having the `<nonSpatiallyDefinable>` tag set to "true" (see [Section 10.6](#)).

The definition of the spatial information type is the following:

```
<!-- SPATIAL INFORMATION TYPE -->
<xs:complexType name="spatialInformationType">
  <xs:sequence>
    <xs:element name="capturePoint" type="capturePointType" minOccurs="0"/>
    <xs:element name="captureArea" type="captureAreaType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

The `<capturePoint>` contains the coordinates of the capture device that is taking the capture, as well as, optionally, the pointing direction (see [Section 10.5.1](#)).

The `<captureArea>` is an optional field containing four points defining the captured area covered by the capture (see [Section 10.5.2](#)).

[10.5.1](#). `<capturePoint>`

The `<capturePoint>` element is used to represent the position and optionally the line of capture of a capture device. `<capturePoint>` MUST be included in spatially definable audio captures, while is optional for spatially definable video captures.

The XML Schema definition of the `<capturePoint>` element type is the following:


```
<!-- POINT TYPE -->
<xs:complexType name="pointType">
  <xs:sequence>
    <xs:element name="x" type="xs:decimal"/>
    <xs:element name="y" type="xs:decimal"/>
    <xs:element name="z" type="xs:decimal"/>
  </xs:sequence>
</xs:complexType>

<!-- CAPTURE POINT TYPE -->
<xs:complexType name="capturePointType">
  <xs:complexContent>
    <xs:extension base="pointType">
      <xs:sequence>
        <xs:element name="lineOfCapturePoint" type="tns:pointType"
          minOccurs="0"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The point type contains three spatial coordinates (x,y,z) representing a point in the space associated with a certain capture scene.

The capture point type extends the point type, i.e., it is represented by three coordinates identifying the position of the capture device, but can add further information. Such further information is conveyed by the `<lineOfCapturePoint>`, which is another point-type element representing the "point on line of capture", that gives the pointing direction of the capture device.

The coordinates of the point on line of capture MUST NOT be identical to the capture point coordinates. For a spatially definable video capture, if the point on line of capture is provided, it MUST belong to the region between the point of capture and the capture area. For a spatially definable audio capture, if the point on line of capture is not provided, the sensitivity pattern should be considered omnidirectional.

10.5.2. <captureArea>

`<captureArea>` is an optional element that can be contained within the spatial information associated with a media capture. It represents the spatial area captured by the media capture. `<captureArea>` MUST be included in the spatial information of spatially definable video

capture, while MUST NOT be associated to audio captures.

The XML representation of that area is provided through a set of four point-type element, <bottomLeft>, <bottomRight>, <topLeft>, and <topRight>, as it can be seen from the following definition:

```
<!-- CAPTURE AREA TYPE -->
<xs:complexType name="captureAreaType">
  <xs:sequence>
    <xs:element name="bottomLeft" type="pointType"/>
    <xs:element name="bottomRight" type="pointType"/>
    <xs:element name="topLeft" type="pointType"/>
    <xs:element name="topRight" type="pointType"/>
  </xs:sequence>
</xs:complexType>
```

<bottomLeft>, <bottomRight>, <topLeft>, and <topRight> MUST be coplanar.

10.6. <nonSpatiallyDefinable>

When media captures are non spatially definable, they are marked with the boolean <nonSpatiallyDefinable> element set to "true" and no <spatialInformation> is provided. Indeed, <nonSpatiallyDefinable> and <spatialInformation> are mutually exclusive tag, according to the <choice> section within the XML Schema definition of the media capture type.

10.7. <content>

A media capture can be (i) an individual media capture or (ii) a multiple content capture (MCC). A multiple content capture is made by different captures that can be arranged spatially (by a composition operation), or temporally (by a switching operation), or that can result from the orchestration of both the techniques. If a media capture is a MCC, then it can show in its XML data model representation the <content> element. It is composed by a list of media capture identifiers ("captureIDREF") and capture scene view identifiers ("sceneViewIDREF"), where the last ones are used as shortcuts to refer to multiple capture identifiers. The referenced captures are used to create the MCC according to a certain strategy. If the <content> element does not appear in a MCC, or it has no child elements, then the MCC is assumed to be made by multiple sources but no information regarding those sources is given.


```
<!-- CONTENT TYPE -->
<xs:complexType name="contentType">
  <xs:sequence>
    <xs:element name="captureIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="sceneViewIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

[10.8.](#) **<synchronizationID>**

<synchronizationID> is an optional element for multiple content captures that contains a numeric identifier. Multiple content captures marked with the same identifier in the <synchronizationID> contain at each time captures coming from the same source. It is the MP that determines what the source for the captures is. By this way, the MP can choose how to group together single captures for the purpose of keeping them synchronized according to the SynchronisationID attribute.

[10.9.](#) **<policy>**

<policy> is an optional element that can be used only for multiple content captures. It indicates the criteria applied to build the multiple content capture using the media captures referenced in <contentCaptureIDs>. Such element can assume a list of pre-defined values ([todo]).

[10.10.](#) **<maxCaptures>**

<maxCaptures> is an optional element that can be used only for multiple content captures. It provides information about the number of media captures that can be represented in the multiple content capture at a time. The type definition is provided below.

```
<!-- MAX CAPTURES TYPE -->
<xs:complexType name="maxCapturesType">
  <xs:simpleContent>
    <xs:extension base="xs:unsignedInt">
      <xs:attribute name="exactNumber" type="xs:boolean"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```



```
</xs:simpleContent>
</xs:complexType>
```

When the "exactNumber" attribute is set to "1", it means the <maxCaptures> element carries the exact number of the media captures appearing at a time. Otherwise, the number of the represented media captures MUST be considered "<=" of the <maxCaptures> value.

[10.11.](#) <individual>

<individual> is a boolean element that MUST be used for single-content captures. Its value is fixed and set to "true". Such element indicates the capture that is being described is not a multiple content capture. Indeed, <individual> and the aforementioned tags related to MCC attributes (from [Section 10.7](#) to [Section 10.10](#)) are mutually exclusive, according to the <choice> section within the XML Schema definition of the media capture type.

[10.12.](#) <description>

<description> is used to provide optionally human-readable textual information about a media capture. The same element is exploited to describe, besides media captures, capture scenes and capture scene views, as it is included in their XML representation. A media capture can be described by using multiple <description> elements, each one providing information in a different language. The <description> element definition is the following:

```
<!-- DESCRIPTION element -->
<xs:element name="description">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="lang" type="xs:language"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

As it can be seen, <description> is a string element with an attribute ("lang") indicating the language used in the textual description.

[10.13.](#) `<priority>`

`<priority>` is an optional unsigned integer field indicating the importance of a media capture according to the Media Provider's perspective. It can be used on the receiver's side to automatically identify the most relevant contribution from the Media Provider. The higher the importance, the lower the contained value. When media captures are marked with a "0" priority value, it means that they are "not subject to priority".

[10.14.](#) `<lang>`

`<lang>` is an optional element containing the language used in the capture, if any.

[10.15.](#) `<mobility>`

`<mobility>` is an optional element indicating whether or not the capture device originating the capture may move during the telepresence session. That optional element can assume one of the three following values: (i) static, (ii) dynamic or (iii) highly dynamic.

[10.16.](#) `<maxCaptureEncodings>`

The optional `<maxCaptureEncodings>` contains an unsigned integer indicating the maximum number of capture encodings that can be simultaneously active for the media capture. If absent, this parameter defaults to 1. The minimum value for this attribute is 1. The number of simultaneous capture encodings is also limited by the restrictions of the encoding group the media capture refers to by means of the `<encGroupIDREF>` element.

[10.17.](#) `<relatedTo>`

The optional `<relatedTo>` element contains the value of the ID attribute of the media capture it refers to. The media capture marked with a `<relatedTo>` element can be for example the translation of a main media capture in a different language.

[10.18.](#) `<view>`

The `<view>` element is an optional tag describing what is represented in the spatial area covered by a media capture. The current possible values are: "table", "lectern", "individual", and "audience", as listed in the enumerative view type in the following.

10.19. <presentation>

The <presentation> element is an optional tag used for media captures conveying information about presentations within the telepresence session. The current possible values are "slides" and "images", as listed in the enumerative presentation type in the following.

10.20. <capturedPeople>

This optional element is used to indicate which telepresence session participants are represented within the media captures. For each participant, a <personIDREF> element is provided.

10.20.1. <personIDREF>

<personIDREF> contains the identifier of the represented person. Metadata about the represented participant can be retrieved by accessing the <people> list ([Section 20](#)).

11. Audio captures

Audio captures inherit all the features of a generic media capture and present further audio-specific characteristics. The XML Schema definition of the audio capture type is reported below:

```
<!-- AUDIO CAPTURE TYPE -->
<xs:complexType name="audioCaptureType">
  <xs:complexContent>
    <xs:extension base="tns:mediaCaptureType">
      <xs:sequence>
        <xs:element name="sensitivityPattern" type="sensitivityPatternType"
          minOccurs="0"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

An example of audio-specific information that can be included is represented by the <sensitivityPattern> element. ([Section 11.1](#)).

11.1. <sensitivityPattern>

The <sensitivityPattern> element is an optional field describing the characteristic of the nominal sensitivity pattern of the microphone capturing the audio signal.

The XML Schema definition is provided below:

```
<xs:simpleType name="sensitivPatternType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="uni"/>
    <xs:enumeration value="shotgun"/>
    <xs:enumeration value="omni"/>
    <xs:enumeration value="figure8"/>
    <xs:enumeration value="cardioid"/>
    <xs:enumeration value="hyper-cardioid"/>
  </xs:restriction>
</xs:simpleType>
```

12. Video captures

Video captures, similarly to audio captures, extend the information of a generic media capture with video-specific features, such as <embeddedText> ([Section 12.1](#)).

The XML Schema representation of the video capture type is provided in the following:

```
<!-- VIDEO CAPTURE TYPE -->
<xs:complexType name="videoCaptureType">
  <xs:complexContent>
    <xs:extension base="tns:mediaCaptureType">
      <xs:sequence>
        <xs:element ref="embeddedText" minOccurs="0"/>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```


12.1. <embeddedText>

The <embeddedText> element is a boolean element indicating that there is text embedded in the video capture. The language used in such embedded textual description is reported in <embeddedText> "lang" attribute.

The XML Schema definition of the <embeddedText> element is:

```
<!-- EMBEDDED TEXT ELEMENT -->
<xs:element name="embeddedText">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:boolean">
        <xs:attribute name="lang" type="xs:language"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

13. Text captures

Also text captures can be described by extending the generic media capture information, similarly to audio captures and video captures.

The XML Schema representation of the text capture type is currently lacking text-specific information, as it can be seen by looking at the definition below:

```
<!-- TEXT CAPTURE TYPE -->
<xs:complexType name="textCaptureType">
  <xs:complexContent>
    <xs:extension base="tns:mediaCaptureType">
      <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```


14. Other capture types

Other media capture types can be described by using the CLUE data model. They can be represented by exploiting "otherCaptureType" type. This media capture type is conceived to be filled with elements defined within extensions of the current schema, i.e., with elements defined in other XML schemas (see [Section 23](#) for an example). The otherCaptureType inherits all the features envisioned for the abstract mediaCaptureType.

The XML Schema representation of the otherCaptureType is the following:

```
<!-- OTHER CAPTURE TYPE -->
<xs:complexType name="otherCaptureType">
  <xs:complexContent>
    <xs:extension base="tns:mediaCaptureType">
      <xs:sequence>
        <xs:any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

15. <captureScene>

A Media Provider organizes the available capture in capture scenes in order to help the receiver both in the rendering and in the selection of the group of captures. Capture scenes are made of media captures and capture scene views, that are set of media captures of the same media type. Each capture scene view is an alternative to represent completely a capture scene for a fixed media type.

The XML Schema representation of a <captureScene> element is the following:


```
<!-- CAPTURE SCENE TYPE -->
<xs:complexType name="captureSceneType">
  <xs:sequence>
    <xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="sceneInformation" type="xcard:vcardType" minOccurs="0"/>
    <xs:element name="sceneViews" type="sceneViewsType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="sceneID" type="xs:ID" use="required"/>
  <xs:attribute name="scale" type="scaleType" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

Each capture scene is identified by a "sceneID" attribute. The <captureScene> element can contain zero or more textual <description> elements, defined as in [Section 10.12](#). Besides <description>, there is the optional <sceneInformation> element ([Section 15.1](#)), which contains structured information about the scene in the vcard format, and the optional <sceneViews> element ([Section 15.2](#)), which is the list of the capture scene views. When no <sceneViews> is provided, the capture scene is assumed to be made by all the media captures showing the value of its sceneID attribute in their mandatory captureSceneIDREF attribute.

[15.1.](#) <sceneInformation>

The <sceneInformation> element contains optional information about the capture scene according to the vcard format.

[15.2.](#) <sceneViews>

The <sceneViews> element is a mandatory field of a capture scene containing the list of scene views. Each scene view is represented by a <sceneView> element ([Section 16](#)).

```
<!-- SCENE VIEWS TYPE -->
<!-- envelope of scene views of a capture scene -->
<xs:complexType name="sceneViewsType">
  <xs:sequence>
    <xs:element name="sceneView" type="sceneViewType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```


15.3. sceneID attribute

The sceneID attribute is a mandatory attribute containing the identifier of the capture scene.

15.4. scale attribute

The scale attribute is a mandatory attribute that specifies the scale of the coordinates provided in the spatial information of the media capture belonging to the considered capture scene. The scale attribute can assume three different values:

"mm" - the scale is in millimeters. Systems which know their physical dimensions (for example professionally installed telepresence room systems) should always provide those real-world measurements.

"unknown" - the scale is not necessarily millimeters, but the scale is the same for every media capture in the capture scene. Systems which don't know specific physical dimensions but still know relative distances should select "unknown" in the scale attribute of the capture scene to be described.

"noscale" - there is no a common physical scale among the media captures of the capture scene. That means the scale could be different for each media capture.

```
<!-- SCALE TYPE -->
<xs:simpleType name="scaleType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="mm"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="noscale"/>
  </xs:restriction>
</xs:simpleType>
```

16. <sceneView>

A <sceneView> element represents a capture scene view, which contains a set of media capture of the same media type describing a capture scene.

A <sceneView> element is characterized as follows.


```
<!-- SCENE VIEW TYPE -->
<xs:complexType name="sceneViewType">
  <xs:sequence>
    <xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="mediaCaptureIDs" type="captureIDListType"/>
  </xs:sequence>
  <xs:attribute name="sceneViewID" type="xs:ID" use="required"/>
</xs:complexType>
```

One or more optional <description> elements provide human-readable information about what the scene view contains. <description> is defined as already seen in [Section 10.12](#).

The remaining child elements are described in the following subsections.

[16.1.](#) <mediaCaptureIDs>

The <mediaCaptureIDs> is the list of the identifiers of the media captures included in the scene view. It is an element of the captureIDListType type, which is defined as a sequence of <captureIDREF> each one containing the identifier of a media capture listed within the <mediaCaptures> element:

```
<!-- CAPTURE ID LIST TYPE -->
<xs:complexType name="captureIDListType">
  <xs:sequence>
    <xs:element name="captureIDREF" type="xs:IDREF"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

[16.2.](#) sceneViewID attribute

The sceneViewID attribute is a mandatory attribute containing the identifier of the capture scene view represented by the <sceneView> element.

[17.](#) <encodingGroup>

The <encodingGroup> element represents an encoding group, which is made by a set of one or more individual encodings and some parameters that apply to the group as a whole. Encoding groups contain references to individual encodings that can be applied to media

captures. The definition of the <encodingGroup> element is the following:

```
<!-- ENCODING GROUP TYPE -->
<xs:complexType name="encodingGroupType">
  <xs:sequence>
    <xs:element name="maxGroupBandwidth" type="xs:unsignedLong"/>
    <xs:element name="encodingIDList" type="encodingIDListType"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="encodingGroupID" type="xs:ID" use="required"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

In the following, the contained elements are further described.

[17.1.](#) <maxGroupBandwidth>

<maxGroupBandwidth> is an optional field containing the maximum bitrate expressed in bit per second that can be shared by the individual encodings included in the encoding group.

[17.2.](#) <encodingIDList>

<encodingIDList> is the list of the individual encoding grouped together in the encoding group. Each individual encoding is represented through its identifier contained within an <encID> element.

```
<!-- ENCODING ID LIST TYPE -->
<xs:complexType name="encodingIDListType">
  <xs:sequence>
    <xs:element name="encID" type="xs:IDREF" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

[17.3.](#) encodingGroupID attribute

The encodingGroupID attribute contains the identifier of the encoding group.

18. <simultaneousSet>

<simultaneousSet> represents a simultaneous transmission set, i.e., a list of captures of the same media type that can be transmitted at the same time by a Media Provider. There are different simultaneous transmission sets for each media type.

```
<!-- SIMULTANEOUS SET TYPE -->
<xs:complexType name="simultaneousSetType">
  <xs:sequence>
    <xs:element name="mediaCaptureIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="sceneViewIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="captureSceneIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="setID" type="xs:ID" use="required"/>
  <xs:attribute name="mediaType" type="xs:string"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

Besides the identifiers of the captures (<mediaCaptureIDREF> elements), also the identifiers of capture scene views and of capture scene can be exploited, as shortcuts (<sceneViewIDREF> and <captureSceneIDREF> elements).

18.1. setID attribute

The "setID" attribute is a mandatory field containing the identifier of the simultaneous set.

When only capture scene identifiers are listed within a simultaneous set, the media type attribute MUST be used in order to determine which media captures can be simultaneously sent together.

18.2. mediaType attribute

The "mediaType" attribute is an optional attribute containing the media type of the captures referenced by the simultaneous set.

When only capture scene identifiers are listed within a simultaneous set, the media type attribute MUST appear in the XML description in

order to determine which media captures can be simultaneously sent together.

[18.3.](#) **<mediaCaptureIDREF>**

<mediaCaptureIDREF> contains the identifier of the media capture that belongs to the simultaneous set.

[18.4.](#) **<sceneViewIDREF>**

<sceneViewIDREF> contains the identifier of the scene view containing a group of capture that are able to be sent simultaneously with the other captures of the simultaneous set.

[18.5.](#) **<captureSceneIDREF>**

<captureSceneIDREF> contains the identifier of the capture scene where all the included captures of a certain media type are able to be sent together with the other captures of the simultaneous set.

[19.](#) **<globalView>**

<globalView> is a set of captures of the same media type representing a summary of the complete Media Provider's offer. The content of a global view is expressed by leveraging only scene view identifiers, put within <sceneViewIDREF> elements. Each global view is identified by a unique identifier within the "globalViewID" attribute.

```
<!-- GLOBAL VIEW TYPE -->
<xs:complexType name="globalViewType">
  <xs:sequence>
    <xs:element name="sceneViewIDREF" type="xs:IDREF"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="globalViewID" type="xs:ID"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```

[20.](#) **<people>**

Information about the participants that are represented in the media captures is conveyed via the <people> element. As it can be seen from the XML Schema depicted below, for each participant, a <person>

element is provided.

```
<!-- PEOPLE TYPE -->
<xs:complexType name="peopleType">
  <xs:sequence>
    <xs:element name="person" type="personType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

  <!-- PERSON TYPE -->
  <xs:complexType name="personType">
    <xs:sequence>
      <xs:element name="personInfo" type="xcard:vcardType" maxOccurs="1"
        minOccurs="0"/>
      <xs:element name="personType" type="personTypeType"
        minOccurs="0"
        maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="personID" type="xs:ID" use="required"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>

<!-- PERSON TYPE TYPE -->
<xs:simpleType name="personTypeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="chairman"/>
    <xs:enumeration value="vice-chairman"/>
    <xs:enumeration value="minute taker"/>
    <xs:enumeration value="presenter"/>
    <xs:enumeration value="translator"/>
    <xs:enumeration value="timekeeper"/>
    <xs:enumeration value="attendee"/>
  </xs:restriction>
</xs:simpleType>
```

20.1. <person>

<person> includes all the metadata related to a person represented within one or more media captures. Such element provides the vcard of the subject (via the <personInfo> element, see [Section 20.1.2](#)) and his conference role(s) (via one or more <personType> elements, see [Section 20.1.3](#)). Furthermore, it has a mandatory "personID"

attribute ([Section 20.1.1](#)).

[20.1.1.](#) **personID attribute**

The "personID" attribute carries the identifier of a represented person. Such identifier can be used to refer to the participant, as in the <capturedPeople> element in media captures representation ([Section 10.20](#)).

[20.1.2.](#) **<personInfo>**

The <personInfo> element is the XML representation of all the fields composing a vcard as specified in the Xcard RFC [[RFC6351](#)]. The vcardType is imported by the Xcard XML Schema provided by [[I-D.ietf-ecrit-additional-data](#)]. As such schema specifies, the <fn> element within <vcard> is mandatory.

[20.1.3.](#) **<personType>**

The value of the <personType> element determines the role of the represented participant within the telepresence session organization. It can be one of the following terms, that are defined in the framework document: "presenter", "timekeeper", "attendee", "minute taker", "translator", "chairman", "vice-chairman".

A participant can have more than one conference role. In that case, more than one <personType> element will appear in his description.

[21.](#) **<captureEncoding>**

A <captureEncoding> is given from the association of a media capture and an individual encoding, to form a capture stream as defined in [[I-D.ietf-clue-framework](#)]. The model of such an entity is provided in the following.

```
<!-- CAPTURE ENCODING TYPE -->
<xs:complexType name="captureEncodingType">
  <xs:sequence>
    <xs:element name="captureID" type="xs:string"/>
    <xs:element name="encodingID" type="xs:string"/>
    <xs:element name="configuredContent" type="contentType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID" use="required"/>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
```


</xs:complexType>

21.1. <captureID>

<captureID> is the mandatory element containing the identifier of the media capture that has been encoded to form the capture encoding.

21.2. <encodingID>

<encodingID> is the mandatory element containing the identifier of the applied individual encoding.

21.3. <configuredContent>

<configuredContent> is an optional element to be used in case of configuration of MCCs. It contains the list of capture identifiers and capture scene view identifiers the Media Consumer wants within the MCC. That element is structured as the <content> element used to describe the content of a MCC, i.e., it contains The total number of the media captures listed in the <configuredContent> must be lower than or equal to the value carried within the <maxCaptures> attribute of the MCC.

22. <clueInfo>

The <clueInfo> element has been left within the XML Schema for representing a drafty version of the body of an ADVERTISEMENT message (see the example section).


```
<!-- CLUE INFO ELEMENT -->
<!-- the <clueInfo> envelope can be seen
      as the ancestor of an <advertisement> envelope -->
<xs:element name="clueInfo" type="clueInfoType"/>

<!-- CLUE INFO TYPE -->
<xs:complexType name="clueInfoType">
  <xs:sequence>
    <xs:element ref="mediaCaptures"/>
    <xs:element ref="encodingGroups"/>
    <xs:element ref="captureScenes"/>
    <xs:element ref="simultaneousSets" minOccurs="0"/>
    <xs:element ref="globalViews" minOccurs="0"/>
    <xs:element ref="people" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="clueInfoID" type="xs:ID" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

23. XML Schema extensibility

The telepresence data model defined in this document is meant to be extensible. Extensions are accomplished by defining elements or attributes qualified by namespaces other than "urn:ietf:params:xml:ns:clue-info" and "urn:ietf:params:xml:ns:vccard-4.0" for use wherever the schema allows such extensions (i.e., where the XML Schema definition specifies "anyAttribute" or "anyElement"). Elements or attributes from unknown namespaces MUST be ignored.

23.1. Example of extension

When extending the CLUE data model, a new schema with a new namespace associated with it needs to be specified.

In the following, an example of extension is provided. The extension defines a new audio capture attribute ("newAudioFeature") and an attribute for characterizing the captures belonging to an "otherCaptureType" defined by the user. An XML document compliant with the extension is also included. The XML file results validated against the current CLUE data model schema.


```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema
  targetNamespace="urn:ietf:params:xml:ns:clue-info-ext"
  xmlns:tns="urn:ietf:params:xml:ns:clue-info-ext"
  xmlns:clue-ext="urn:ietf:params:xml:ns:clue-info-ext"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:clue-info-ext"
  xmlns:xcard="urn:ietf:params:xml:ns:vcard-4.0"
  xmlns:info="urn:ietf:params:xml:ns:clue-info"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <!-- Import xcard XML schema -->
  <xs:import namespace="urn:ietf:params:xml:ns:vcard-4.0"
    schemaLocation="xcard.xsd"/>

  <!-- Import CLUE XML schema -->
  <xs:import namespace="urn:ietf:params:xml:ns:clue-info"
    schemaLocation="data-model-schema-07.xsd"/>

  <!-- ELEMENT DEFINITIONS -->
  <xs:element name="newAudioFeature" type="xs:string"/>
  <xs:element name="otherMediaCaptureTypeFeature" type="xs:string"/>

</xs:schema>
```



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<clueInfo xmlns="urn:ietf:params:xml:ns:clue-info"
xmlns:ns2="urn:ietf:params:xml:ns:vcard-4.0"
xmlns:ns3="urn:ietf:params:xml:ns:clue-info-ext"
clueInfoID="NapoliRoom">
  <mediaCaptures>
    <mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="audioCaptureType"
captureID="AC0"
mediaType="audio">
      <captureSceneIDREF>CS1</captureSceneIDREF>
      <nonSpatiallyDefinable>true</nonSpatiallyDefinable>
      <individual>true</individual>
      <encGroupIDREF>EG1</encGroupIDREF>
      <ns3:newAudioFeature>newAudioFeatureValue</ns3:newAudioFeature>
    </mediaCapture>
    <mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="otherCaptureType"
captureID="OMC0"
mediaType="other media type">
      <captureSceneIDREF>CS1</captureSceneIDREF>
      <nonSpatiallyDefinable>true</nonSpatiallyDefinable>
      <encGroupIDREF>EG1</encGroupIDREF>
      <ns3:otherMediaCaptureTypeFeature>OtherValue
      </ns3:otherMediaCaptureTypeFeature>
    </mediaCapture>
  </mediaCaptures>
  <encodingGroups>
    <encodingGroup encodingGroupID="EG1">
      <maxGroupBandwidth>300000</maxGroupBandwidth>
      <encodingIDList>
        <encID>ENC4</encID>
        <encID>ENC5</encID>
      </encodingIDList>
    </encodingGroup>
  </encodingGroups>
  <captureScenes>
    <captureScene scale="unknown" sceneID="CS1"/>
  </captureScenes>
</clueInfo>
```

24. Security considerations

This document defines an XML Schema data model for telepresence scenarios. The modeled information is identified in the CLUE framework as the needed one in order to enable a full-optional media stream negotiation and rendering. Indeed, the XML elements herein

defined are used within CLUE protocol messages to describe both the media streams representing the MP's telepresence offer and the desired selection requested by the MC. Security concerns described in [[I-D.ietf-clue-framework](#)], Section 15, apply to this document.

Data model information carried within CLUE messages SHOULD be accessed only by authenticated endpoints. Indeed, some information published by the MP might reveal sensitive data about who and what is represented in the transmitted streams. The vCards included in the <personInfo> elements ([Section 20.1](#)) mandatorily contains the identity of the represented person. Optionally vCards can also carry the person's contact addresses, together with his/her photo and other personal data. Similar privacy-critical information can be conveyed by means of <sceneInformation> elements ([Section 15.1](#)) describing the capture scenes. The <description> elements also can specify details that should be protected about the content of media captures ([Section 10.12](#)), capture scenes ([Section 15](#)), scene views ([Section 16](#)).

Integrity attacks to the data model information incapsulated in CLUE messages can invalidate the success of the telepresence session's setup by misleading the MC's and MP's interpretation of the offered and desired media streams.

The assurance of the authenticated access and of the integrity of the data model information is up to the involved transport mechanisms, namely the CLUE protocol [[I-D.ietf-clue-protocol](#)] and the CLUE data channel [[I-D.ietf-clue-datachannel](#)].

[25.](#) IANA considerations

ToDo.

[25.1.](#) XML Schema registration

ToDo.

[25.2.](#) XML namespace registration

ToDo.

[26.](#) Sample XML file

The following XML document represents a schema compliant example of a CLUE telepresence scenario. Taking inspiration from the examples described in the framework draft ([[I-D.ietf-clue-framework](#)]), it is provided the XML representation of an endpoint-style Media Provider's offer.

There are three cameras, where the central one is also able of capturing a zoomed-out view of the overall telepresence room. Besides the three video captures coming from such cameras, the MP makes available a further multi-content capture about the loudest segment of the room, obtained by switching the video source across the three cameras. For the sake of simplicity, only one audio capture is advertised for the audio of the whole room.

The three cameras are placed in front of three participants (Alice, Bob and Ciccio), whose vcard and conference roles details are also provided.

Media captures are arranged into four capture scene views:

1. (VC0, VC1, VC2) - left, center and right camera video captures
2. (VC3) - video capture associated with loudest room segment
3. (VC4) - video capture zoomed out view of all people in the room
4. (AC0) - main audio

There are two encoding groups: (i) EG0, for video encodings, and (ii) EG1, for audio encodings.

As to the simultaneous sets, only VC1 and VC4 cannot be transmitted simultaneously since they are captured by the same device, i.e., the central camera (VC4 is a zoomed-out view while VC1 is a focused view of the front participant). The simultaneous sets would then be the following:

SS1 made by VC3 and all the captures in the first capture scene view (VC0,VC1,VC2);

SS2 made by VC3, VC0, VC2, VC4

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<clueInfo xmlns="urn:ietf:params:xml:ns:clue-info"
  xmlns:ns2="urn:ietf:params:xml:ns:vcard-4.0" clueInfoID="NapoliRoom">
  <mediaCaptures>
    <mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="audioCaptureType" captureID="AC0" mediaType="audio">
      <captureSceneIDREF>CS1</captureSceneIDREF>
      <encGroupIDREF>EG1</encGroupIDREF>
      <spatialInformation>
```



```
<capturePoint>
  <x>0.5</x>
  <y>1.0</y>
  <z>0.5</z>
  <lineOfCapturePoint>
    <x>0.5</x>
    <y>0.0</y>
    <z>0.5</z>
  </lineOfCapturePoint>
</capturePoint>
</spatialInformation>
<individual>true</individual>
<description lang="en">main audio from the room</description>
<priority>1</priority>
<lang>it</lang>
<mobility>static</mobility>
<view>room</view>
<capturedPeople>
  <personIDREF>alice</personIDREF>
  <personIDREF>bob</personIDREF>
  <personIDREF>ciccio</personIDREF>
</capturedPeople>
<maxCaptureEncodings>1</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="videoCaptureType" captureID="VC0" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <spatialInformation>
    <capturePoint>
      <x>0.5</x>
      <y>1.0</y>
      <z>0.5</z>
      <lineOfCapturePoint>
        <x>0.5</x>
        <y>0.0</y>
        <z>0.5</z>
      </lineOfCapturePoint>
    </capturePoint>
  </spatialInformation>
  <individual>true</individual>
  <description lang="en">left camera video capture</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <capturedPeople>
    <personIDREF>ciccio</personIDREF>
```



```
</capturedPeople>
  <maxCaptureEncodings>2</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC1" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <spatialInformation>
    <capturePoint>
      <x>0.5</x>
      <y>1.0</y>
      <z>0.5</z>
      <lineOfCapturePoint>
        <x>0.5</x>
        <y>0.0</y>
        <z>0.5</z>
      </lineOfCapturePoint>
    </capturePoint>
  </spatialInformation>
  <individual>true</individual>
  <description lang="en">central camera video capture</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <capturedPeople>
    <personIDREF>alice</personIDREF>
  </capturedPeople>
  <maxCaptureEncodings>2</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC2" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <spatialInformation>
    <capturePoint>
      <x>0.5</x>
      <y>1.0</y>
      <z>0.5</z>
      <lineOfCapturePoint>
        <x>0.5</x>
        <y>0.0</y>
        <z>0.5</z>
      </lineOfCapturePoint>
    </capturePoint>
  </spatialInformation>
  <individual>true</individual>
  <description lang="en">right camera video capture</description>
```



```
<priority>1</priority>
<lang>it</lang>
<mobility>static</mobility>
<view>individual</view>
<capturedPeople>
  <personIDREF>bob</personIDREF>
</capturedPeople>
<maxCaptureEncodings>2</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC3" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <nonSpatiallyDefinable>true</nonSpatiallyDefinable>
  <policy>Soundlevel:0</policy>
  <description lang="en">loudest room segment</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <maxCaptureEncodings>1</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC4" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <spatialInformation>
    <capturePoint>
      <x>0.5</x>
      <y>1.0</y>
      <z>0.5</z>
      <lineOfCapturePoint>
        <x>0.5</x>
        <y>0.0</y>
        <z>0.5</z>
      </lineOfCapturePoint>
    </capturePoint>
  </spatialInformation>
  <individual>true</individual>
  <description lang="en">zoomed out view of all people in the
room</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>room</view>
  <capturedPeople>
    <personIDREF>alice</personIDREF>
    <personIDREF>bob</personIDREF>
```



```
        <personIDREF>ciccio</personIDREF>
      </capturedPeople>
      <maxCaptureEncodings>1</maxCaptureEncodings>
    </mediaCapture>
  </mediaCaptures>
  <encodingGroups>
    <encodingGroup encodingGroupID="EG0">
      <maxGroupBandwidth>600000</maxGroupBandwidth>
      <encodingIDList>
        <encID>ENC1</encID>
        <encID>ENC2</encID>
        <encID>ENC3</encID>
      </encodingIDList>
    </encodingGroup>
    <encodingGroup encodingGroupID="EG1">
      <maxGroupBandwidth>300000</maxGroupBandwidth>
      <encodingIDList>
        <encID>ENC4</encID>
        <encID>ENC5</encID>
      </encodingIDList>
    </encodingGroup>
  </encodingGroups>
  <captureScenes>
    <captureScene scale="unknown" sceneID="CS1">
      <sceneViews>
        <sceneView sceneViewID="SE1">
          <mediaCaptureIDs>
            <captureIDREF>VC0</captureIDREF>
            <captureIDREF>VC1</captureIDREF>
            <captureIDREF>VC2</captureIDREF>
          </mediaCaptureIDs>
        </sceneView>
        <sceneView sceneViewID="SE2">
          <mediaCaptureIDs>
            <captureIDREF>VC3</captureIDREF>
          </mediaCaptureIDs>
        </sceneView>
        <sceneView sceneViewID="SE3">
          <mediaCaptureIDs>
            <captureIDREF>VC4</captureIDREF>
          </mediaCaptureIDs>
        </sceneView>
        <sceneView sceneViewID="SE4">
          <mediaCaptureIDs>
            <captureIDREF>VC4</captureIDREF>
          </mediaCaptureIDs>
        </sceneView>
      </sceneViews>
```



```
    </captureScene>
  </captureScenes>
  <simultaneousSets>
    <simultaneousSet setID="SS1">
      <mediaCaptureIDREF>VC3</mediaCaptureIDREF>
      <sceneViewIDREF>SE1</sceneViewIDREF>
    </simultaneousSet>
    <simultaneousSet setID="SS2">
      <mediaCaptureIDREF>VC0</mediaCaptureIDREF>
      <mediaCaptureIDREF>VC2</mediaCaptureIDREF>
      <mediaCaptureIDREF>VC4</mediaCaptureIDREF>
      <mediaCaptureIDREF>VC3</mediaCaptureIDREF>
    </simultaneousSet>
  </simultaneousSets>
  <people>
    <person personID="bob">
      <personInfo>
        <ns2:fn>
          <ns2:text>Bob</ns2:text>
        </ns2:fn>
      </personInfo>
      <personType>minute taker</personType>
    </person>
    <person personID="alice">
      <personInfo>
        <ns2:fn>
          <ns2:text>Alice</ns2:text>
        </ns2:fn>
      </personInfo>
      <personType>presenter</personType>
    </person>
    <person personID="ciccio">
      <personInfo>
        <ns2:fn>
          <ns2:text>Ciccio</ns2:text>
        </ns2:fn>
      </personInfo>
      <personType>chairman</personType>
      <personType>timekeeper</personType>
    </person>
  </people>
</clueInfo>
```


27. MCC example

Enhancing the scenario presented in the previous example, the Media Provider is able to advertise a composed capture VC7 made by a big picture representing the current speaker (VC3) and two picture-in-picture boxes representing the previous speakers (the previous one -VC5- and the oldest one -VC6). The provider does not want to instantiate and send VC5 and VC6, so it does not associate any encoding group with them. Their XML representations are provided for enabling the description of VC7.

A possible description for that scenario could be the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<clueInfo xmlns="urn:ietf:params:xml:ns:clue-info"
xmlns:ns2="urn:ietf:params:xml:ns:vcard-4.0" clueInfoID="NapoliRoom">
  <mediaCaptures>
    <mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="audioCaptureType" captureID="AC0" mediaType="audio">
      <captureSceneIDREF>CS1</captureSceneIDREF>
      <encGroupIDREF>EG1</encGroupIDREF>
      <spatialInformation>
        <capturePoint>
          <x>0.5</x>
          <y>1.0</y>
          <z>0.5</z>
          <lineOfCapturePoint>
            <x>0.5</x>
            <y>0.0</y>
            <z>0.5</z>
          </lineOfCapturePoint>
        </capturePoint>
      </spatialInformation>
      <individual>true</individual>
      <description lang="en">main audio from the room</description>
      <priority>1</priority>
      <lang>it</lang>
      <mobility>static</mobility>
      <view>room</view>
      <capturedPeople>
        <personIDREF>alice</personIDREF>
        <personIDREF>bob</personIDREF>
        <personIDREF>ciccio</personIDREF>
      </capturedPeople>
      <maxCaptureEncodings>1</maxCaptureEncodings>
    </mediaCapture>
  </mediaCaptures>
</clueInfo>
```



```
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC0" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <spatialInformation>
    <capturePoint>
      <x>0.5</x>
      <y>1.0</y>
      <z>0.5</z>
      <lineOfCapturePoint>
        <x>0.5</x>
        <y>0.0</y>
        <z>0.5</z>
      </lineOfCapturePoint>
    </capturePoint>
  </spatialInformation>
  <individual>true</individual>
  <description lang="en">left camera video capture</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <capturedPeople>
    <personIDREF>ciccio</personIDREF>
  </capturedPeople>
  <maxCaptureEncodings>2</maxCaptureEncodings>
</mediaCapture>

<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC1" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <spatialInformation>
    <capturePoint>
      <x>0.5</x>
      <y>1.0</y>
      <z>0.5</z>
      <lineOfCapturePoint>
        <x>0.5</x>
        <y>0.0</y>
        <z>0.5</z>
      </lineOfCapturePoint>
    </capturePoint>
  </spatialInformation>
  <individual>true</individual>
  <description lang="en">central camera video capture</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
```



```
<view>individual</view>
<capturedPeople>
  <personIDREF>alice</personIDREF>
</capturedPeople>
<maxCaptureEncodings>2</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC2" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <spatialInformation>
    <capturePoint>
      <x>0.5</x>
      <y>1.0</y>
      <z>0.5</z>
      <lineOfCapturePoint>
        <x>0.5</x>
        <y>0.0</y>
        <z>0.5</z>
      </lineOfCapturePoint>
    </capturePoint>
  </spatialInformation>
  <individual>true</individual>
  <description lang="en">right camera video capture</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <capturedPeople>
    <personIDREF>bob</personIDREF>
  </capturedPeople>
  <maxCaptureEncodings>2</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC3" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <nonSpatiallyDefinable>true</nonSpatiallyDefinable>
  <content>
    <sceneViewIDREF>SE1</sceneViewIDREF>
  </content>
  <policy>Soundlevel:0</policy>
  <description lang="en">loudest room segment</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <maxCaptureEncodings>1</maxCaptureEncodings>
```



```

</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC4" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <encGroupIDREF>EG0</encGroupIDREF>
  <spatialInformation>
    <capturePoint>
      <x>0.5</x>
      <y>1.0</y>
      <z>0.5</z>
      <lineOfCapturePoint>
        <x>0.5</x>
        <y>0.0</y>
        <z>0.5</z>
      </lineOfCapturePoint>
    </capturePoint>
  </spatialInformation>
  <individual>true</individual>
  <description lang="en">zoomed out view of all people in the room</
description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>room</view>
  <capturedPeople>
    <personIDREF>alice</personIDREF>
    <personIDREF>bob</personIDREF>
    <personIDREF>ciccio</personIDREF>
  </capturedPeople>
  <maxCaptureEncodings>1</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC5" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <nonSpatiallyDefinable>true</nonSpatiallyDefinable>
  <content>
    <sceneViewIDREF>SE1</sceneViewIDREF>
  </content>
  <policy>Soundlevel:1</policy>
  <description lang="en">penultimate loudest room segment</
description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <maxCaptureEncodings>1</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```
xsi:type="videoCaptureType" captureID="VC6" mediaType="video">  
  <captureSceneIDREF>CS1</captureSceneIDREF>
```

```
<nonSpatiallyDefinable>true</nonSpatiallyDefinable>
<content>
  <sceneViewIDREF>SE1</sceneViewIDREF>
</content>
<policy>Soundlevel:2</policy>
<description lang="en">last but two loudest room segment</
description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <maxCaptureEncodings>1</maxCaptureEncodings>
</mediaCapture>
<mediaCapture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="videoCaptureType" captureID="VC7" mediaType="video">
  <captureSceneIDREF>CS1</captureSceneIDREF>
  <nonSpatiallyDefinable>true</nonSpatiallyDefinable>
  <content>
    <captureIDREF>VC3</captureIDREF>
    <captureIDREF>VC5</captureIDREF>
    <captureIDREF>VC6</captureIDREF>
  </content>
  <description lang="en">big picture of the current speaker +
  pips about previous speakers</description>
  <priority>1</priority>
  <lang>it</lang>
  <mobility>static</mobility>
  <view>individual</view>
  <maxCaptureEncodings>1</maxCaptureEncodings>
</mediaCapture>
</mediaCaptures>
<encodingGroups>
  <encodingGroup encodingGroupID="EG0">
    <maxGroupBandwidth>600000</maxGroupBandwidth>
    <encodingIDList>
      <encID>ENC1</encID>
      <encID>ENC2</encID>
      <encID>ENC3</encID>
    </encodingIDList>
  </encodingGroup>
  <encodingGroup encodingGroupID="EG1">
    <maxGroupBandwidth>300000</maxGroupBandwidth>
    <encodingIDList>
      <encID>ENC4</encID>
      <encID>ENC5</encID>
    </encodingIDList>
  </encodingGroup>
</encodingGroups>
```

<captureScenes>

Presta & Romano

Expires April 2, 2015

[Page 54]

```
<captureScene scale="unknown" sceneID="CS1">
  <sceneViews>
    <sceneView sceneViewID="SE1">
      <description lang="en">participants' individual
        videos</description>
      <mediaCaptureIDs>
        <captureIDREF>VC0</captureIDREF>
        <captureIDREF>VC1</captureIDREF>
        <captureIDREF>VC2</captureIDREF>
      </mediaCaptureIDs>
    </sceneView>
    <sceneView sceneViewID="SE2">
      <description lang="en">loudest segment of the
        room</description>
      <mediaCaptureIDs>
        <captureIDREF>VC3</captureIDREF>
      </mediaCaptureIDs>
    </sceneView>
    <sceneView sceneViewID="SE5">
      <description lang="en">loudest segment of the
        room + pips</description>
      <mediaCaptureIDs>
        <captureIDREF>VC7</captureIDREF>
      </mediaCaptureIDs>
    </sceneView>
    <sceneView sceneViewID="SE4">
      <description lang="en">room audio</description>
      <mediaCaptureIDs>
        <captureIDREF>AC0</captureIDREF>
      </mediaCaptureIDs>
    </sceneView>
    <sceneView sceneViewID="SE3">
      <description lang="en">room video</description>
      <mediaCaptureIDs>
        <captureIDREF>VC4</captureIDREF>
      </mediaCaptureIDs>
    </sceneView>
  </sceneViews>
</captureScene>
</captureScenes>
<simultaneousSets>
  <simultaneousSet setID="SS1">
    <mediaCaptureIDREF>VC7</mediaCaptureIDREF>
    <sceneViewIDREF>SE1</sceneViewIDREF>
  </simultaneousSet>
  <simultaneousSet setID="SS2">
    <mediaCaptureIDREF>VC0</mediaCaptureIDREF>
    <mediaCaptureIDREF>VC2</mediaCaptureIDREF>
```



```
        <mediaCaptureIDREF>VC4</mediaCaptureIDREF>
        <mediaCaptureIDREF>VC7</mediaCaptureIDREF>
    </simultaneousSet>
</simultaneousSets>
<people>
  <person personID="bob">
    <personInfo>
      <ns2:fn>
        <ns2:text>Bob</ns2:text>
      </ns2:fn>
    </personInfo>
    <personType>minute taker</personType>
  </person>
  <person personID="alice">
    <personInfo>
      <ns2:fn>
        <ns2:text>Alice</ns2:text>
      </ns2:fn>
    </personInfo>
    <personType>presenter</personType>
  </person>
  <person personID="ciccio">
    <personInfo>
      <ns2:fn>
        <ns2:text>Ciccio</ns2:text>
      </ns2:fn>
    </personInfo>
    <personType>chairman</personType>
    <personType>timekeeper</personType>
  </person>
</people>
</clueInfo>
```

28. Diff with [draft-ietf-clue-data-model-schema-06](#) version

- o Capture Scene Entry/Entries renamed as Capture Scene View/Views in the text, <sceneEntry>/<sceneEntries> renamed as <sceneView>/<sceneViews> in the XML schema.
- o Global Scene Entry/Entries renamed as Global View/Views in the text, <globalSceneEntry>/<globalSceneEntries> renamed as <globalView>/<globalViews>
- o Security section added.

- o Extensibility: a new type is introduced to describe other types of media capture (otherCaptureType), text and example added.
- o Spatial information section updated: capture point optional, text now is coherent with the framework one.
- o Audio capture description: <sensitivityPattern> added, <audioChannelFormat> removed, <captureArea> disallowed.
- o Simultaneous set definition: added <captureSceneIDREF> to refer to capture scene identifiers as shortcuts and an optional mediaType attribute which is mandatory to use when only capture scene identifiers are listed.
- o Encoding groups: removed the constraint of the same media type.
- o Updated text about media captures without <encodingGroupIDREF> (optional in the XML schema).
- o "mediaType" attribute removed from homogeneous groups of capture (scene views and global views)
- o "mediaType" attribute removed from the global view textual description.
- o "millimeters" scale value changed in "mm"

29. Diff with [draft-ietf-clue-data-model-schema-04](#) version

globalCaptureEntries/Entry renamed as globalSceneEntries/Entry;

sceneInformation added;

Only capture scene entry identifiers listed within global scene entries (media capture identifiers removed);

<participants> renamed as <people> in the >clueInfo< template

<vcard> renamed as <personInfo> to synch with the framework terminology

<participantType> renamed as <personType> to synch with the framework terminology

<participantIDs> renamed as <capturedPeople> in the media capture type definition to remove ambiguity

Examples have been updated with the new definitions of <globalSceneEntries> and of <people>.

30. Diff with [draft-ietf-clue-data-model-schema-03](#) version

encodings section has been removed

global capture entries have been introduced

capture scene entry identifiers are used as shortcuts in listing the content of MCC (similarly to simultaneous set and global capture entries)

Examples have been updated. A new example with global capture entries has been added.

<encGroupIDREF> has been made optional.

<single> has been renamed into <individual>

Obsolete comments have been removed.

participants information has been added.

31. Diff with [draft-ietf-clue-data-model-schema-02](#) version

captureParameters and encodingParameters have been removed from the captureEncodingType

data model example has been updated and validated according to the new schema. Further description of the represented scenario has been provided.

A multiple content capture example has been added.

Obsolete comments and references have been removed.

32. Informative References

- | | |
|-----------------------------|--|
| [I-D.ietf-clue-datachannel] | Holmberg, C., "CLUE Protocol Data Channel",
draft-ietf-clue-datachannel-01
(work in progress), September 2014. |
| [I-D.ietf-clue-framework] | Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams",
draft-ietf-clue-framework-17 (work |

in progress), September 2014.

- [I-D.ietf-clue-protocol] Presta, R. and S. Romano, "CLUE protocol",
[draft-ietf-clue-protocol-01](#) (work in progress), June 2014.
- [I-D.ietf-ecrit-additional-data] Rosen, B., Tschofenig, H., Marshall, R., Randy, R., and J. Winterbottom, "Additional Data related to an Emergency Call",
[draft-ietf-ecrit-additional-data-22](#) (work in progress), April 2014.
- [RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description Protocol (SDP) Content Attribute", [RFC 4796](#), February 2007.
- [RFC6351] Perreault, S., "xCard: vCard XML Representation", [RFC 6351](#), August 2011.

Authors' Addresses

Roberta Presta
University of Napoli
Via Claudio 21
Napoli 80125
Italy

EMail: roberta.presta@unina.it

Simon Pietro Romano
University of Napoli
Via Claudio 21
Napoli 80125
Italy

EMail: spromano@unina.it

