Text

Description automatically generatedISO/IEC JTC 1/SC 29/WG 03 N01454

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output document

**Title:** Text of ISO/IEC 23090-14 2nd edition CDAM 1: Support of MPEG-I immersive audio, scene understanding and other extensions

**Date of document:** 2025-

**Source:** WG03

**No. of pages:** (with cover page)

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

*Modify clause 8 with the following content*

*In clause 8.1.1 change*

AR anchoring allows for anchoring of a scene or some root nodes of a scene into the user’s physical environment to provide an AR experience. AR anchoring is supported at the scene level and at the node level, as well as an extension through the definition of the MPEG\_anchor extension.

*To*

AR anchoring allows for anchoring of a scene or some root nodes of a scene into the user’s physical environment to provide an AR experience. AR anchoring is supported at the scene level and at the node level, as well as an extension through the definition of the MPEG\_anchor extension.

Once anchored, the elements of the scene may interact with the physical environment. The anchoring extension also specify how the representation of this environment may be provided to manage these interactions and to reach expected AR experience.

*In clause 8.1.2, change the table 27 to*

Table 27: Definition of the Anchor object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| trackable | integer | M |  | Index of the trackable in the trackables array that will be used for this anchor. |
| requiresAnchoring | boolean | M |  | Indicates if AR anchoring is required for the rendering of the associated nodes.  If TRUE, the application shall skip the virtual assets attached to this anchor until the pose of this anchor in the real world is known.  if FALSE, the application shall process the virtual assets attached to this anchor |
| minimumRequiredSpace | vec3 | O | (0,0,0) | Space required to anchor the AR asset (width, height, depth in meters). This space corresponds to an axis-aligned bounding box, placed at the origin of the trackable space. Width is aligned with x axis, height with y axis and depth with z axis, expressed in trackable local space. This value shall be compared to the bounding box of the real-world available space associated with the trackable as estimated by the XR runtime. |
| aligned | enumeration | O | NOT\_USED | The aligned flag may take one of the following values: NOT\_USED=0, ALIGNED\_NOTSCALED=1, ALIGNED\_SCALED=2.  If ALIGNED\_SCALED is set, the bounding box of the virtual assets attached to that anchor is aligned and scaled to match the bounding box of the real-world available space associated with the trackable as estimated by the XR runtime. |
| actions | array(number) | O | N/A | Indices of the actions in the actions array of the interactivity extension to be executed once the pose of this anchor is determined. An example is a setTransform action to place the virtual assets attached to that anchor. |
| light | integer | O | N/A | Reference to an item in the lights array of the MPEG\_lights\_texture\_based extension. |
| recommendedSpatialComputingConfig | object | O | N/A | Provides a set of recommended parameters specifying the needed spatial description. The semantics of this object is given in Table 8.1- 1. |

*And add the following table after the table 27*

Table 8.1- 1: RecommendedSpatialComputingConfig Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| scanOptions | array | O | N/A | Array of options (enumeration) for the scan computation: possible values are given in Table 8.1- 2. |
| scanDetails | object | O | N/A | Specifies the required level of detail for the representation. The semantics is presented in Table 8.1- 3. |
| scanUpdate | object | O | N/A | Specifies the frequency at which the spatial description must be updated (Table 8.1- 4). |
| scanVolumes | array | O | N/A | Array of bounding volumes that determine the spaces where scanned objects must be provided. Real scan objects that intersect one or more of the bounding volumes should be provided, and all other objects ignored. The semantics for these volumes are presented inTable 8.1- 5. |
| realSemantic | array | O | N/A | Semantic descriptions of nodes that are needed (“table”, “room”, “chair”, “wall”, “light”, “freespace” …) |
| lightOptions | array | O | N/A | Array of option (enumeration) for the light extraction: possible values are given in Table 8.1- 6. |
| lightUpdate | object | O | N/A | Specifies the frequency at which the extraction of real light must be updated (Table 8.1- 4) |

Table 8.1- 2: Possibles values for a scanOptions item

|  |  |
| --- | --- |
| **Enumeration value** | **Description** |
| PLANE = 0 | Request plane data for scanned objects |
| PLANAR\_MESH | Request planar meshes for scanned objects |
| VISUAL\_MESH | Request 3D visualization meshes for scanned objects |
| COLLIDER\_MESH | Request 3D collider meshes for scanned objects |
| FREE\_VOLUME | Request to get the available space around a trackable |
| BOUNDING\_BOX | Request a simplified collider mesh |
| TEXTURED\_MESH | Request mesh with a texture |

Table 8.1- 3: Semantics for the scanDetail object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| primitivesNumber | number | O | N/A | Gives the quantity of geometric primitives per m3 |
| textureOption | enumeration | O | LOW\_RES | Gives the resolution of the texture of textured mesh. Possible values are LOW\_RES (0) or HIGH\_RES(1). |

Table 8.1- 4: Semantics of a *scanUpdate* or *lightUpdate* object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| occurrence | enumeration | O | ONCE | An occurrence value:   * ONCE = 0: the update is performed only once, for instance in case of a static real scene, * N\_FRAME: the update is performed periodically, every N rendering frames, N depending on the dynamism of the real scene. The N value is provided in the *frameNumber* parameter. * AUTO: the update frequency is managed by the module that compute the representation. This module may perform an analysis to detect significant changes in the real world and start an update. This analysis may be performed from raw images data, like RGB images from a camera or depth images from a depth sensor. |
| numberOfFrames | number | O | N/A | Indicate the periodicity, in number of frames, of the update, when the *occurrences* value is N\_FRAME. |

Table 8.1- 5: semantics for a scanVolumes item

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| type | enumeration | M |  | SPHERE=0, BOX, FRUSTUM |
| If (type == SPHERE) |  |  |  |  |
| center | array | M |  | 3D coordinate of the center of the sphere |
| radius | number | M |  | Radius of the sphere in meters. |
| If (type == BOX) |  |  |  |  |
| pose | matrix | M |  | 4\*4 matrix representing the center position and orientation of the box |
| extents | array | O |  | Edge-to-edge length of the box along each dimension. |
| If (type == FRUSTUM) |  |  |  |  |
| pose | matrix | M |  | 4\*4 matrix representing the position and orientation of the tip of the frustum |
| fov | vec4 | M |  | Angles of the four sides of the frustum |
| far | number | M |  | Positive distance of the far plane of the frustum |
| near | number | M |  | Positive distance of the near plane of the frustum |
| 3D coordinates are expressed in the XR space related to trackable associated to the anchor. | | | | |

Table 8.1- 6: Possible values for a lightOptions item

|  |  |
| --- | --- |
| **Enumeration value** | **Description** |
| DirectionalLight = 0 | Request the extraction of directional lights |
| EnvLight | Request the extraction of environment lights |
| PointLight | Request the extraction of point lights |
| SpotLight | Request the extraction of spot lights |
| AreaLight | Request the extraction of area lights |

*In clause 8.1.3 change*

If the array of action is not empty, the actions are executed once the pose of anchor is determined. If the tracking status is set to FALSE after being TRUE, actions are not canceled.

*To*

If the array of action is not empty, the actions are executed once the pose of anchor is determined. If the tracking status is set to FALSE after being TRUE, actions are not canceled.

When processing the MPEG\_anchor extension, if a ***recommendedSpatialComputingConfig*** objectis present, the Presentation Engine checks if it can retrieve the recommended spatial description, specified in the parameters such as scanOptions, scanDetails, scanUpdate, scanVolumes, scanSemantic, lightUpdate and lightOptions.  
If all the recommended parameters are not satisfied, the Presentation Engine may continue the rendering of the scene with the available spatial description, but possibly with a degraded XR experience.

At runtime, the Presentation Engine may then request elements of the spatial description according to the ***recommendedSpatialComputingConfig*** parameters***.***

*In clause 8.2.1 change*

The MPEG\_node\_interactivity extension is used to complement the interactivity defined at the scene level. One particular case is the definition of the parameters for the physics engine. That is, when an MPEG\_node\_interactivity extension contains a trigger of type TRIGGER\_COLLISION without being referenced by a trigger of type TRIGGER\_COLLISION at the MPEG\_scene\_interactivity extension, this node shall not be considered for collision detection and instead only be used by the physics engine.

*To*

The MPEG\_node\_interactivity extension is used to complement the interactivity extension defined at the scene level. One particular case is the definition of the parameters for a physics engine. That is, when an MPEG\_node\_interactivity extension contains a trigger of type TRIGGER\_COLLISION without being referenced by a trigger of type TRIGGER\_COLLISION at the MPEG\_scene\_interactivity extension, this node shall not be considered for collision detection and instead only be used by the physics engine.

When specified, at node and/or scene level, a physics object specifies a set of parameters for the physic engine, used by the application.

*In clause 8.2.2.1, change the table 31 to*

**Table 31: Semantic of the MPEG\_scene\_interactivity extension**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| physics | object | O | N/A | Provides a set of parameters at scene level to be used for the physics simulation. The semantics of this object is given in table xx3. |
| triggers | array | M | [] | Contains the definition of all the triggers used in that scene |
| actions | array | M | [] | Contains the definition of all the actions used in that scene |
| behaviors | array | M | [] | Contains the definition of all the behaviors used in that scene. A behavior is composed of a pair of (triggers, actions), control parameters of triggers and actions, a priority weight and an optional interrupt action |

*And add the following text*

The semantics of a *physics* object at scene level, are defined in Table 8.2- 1.

**Table 8.2- 1: Semantic of a scene level physics object**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| recommendedPhysicsHighPrecision | boolean | O | false | Determines whether the application should enable a more deterministic and precise physic simulation |
| gravity | number | O | -9.81 | Determine the gravity for the whole scene. In meter per secondsquare (m.s-2), as defined in the international unit system. |
| recommendedPhysicsFrameRate | number | O | 50 | Provides the recommended frame rate at which the Physics Engine should operate. In frame per second, as defined in the international unit system. |
| bounceThreshold | number | O | 1 | A contact with a relative velocity below this threshold will not result in a bounce. In meter per second (m.s-1), as defined in the international unit system. |

*In clause 8.2.2.2, change the table 51 to*

**Table 51: Semantic of the MPEG\_node\_interactivity.trigger extension**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| type | enumeration | M |  | One element of Table 8 that defines the type of the trigger. |
| if (type == TRIGGER\_COLLISION){ |  |  |  |  |
| collider | integer | M |  | the index of the mesh element that provides the collider geometry for the current node.  The collider mesh may reference a material. |
| isStatic | boolean | M |  | If True, the collider is defined as a static collider. |
| physics | object | O | N/A | Provides a set of parameters at node level to be used for the physics simulation. The semantics of this object is given in table xx4. |
| primitives | array(Primitive) | O | N/A | List of primitives used to activate the proximity or collision trigger.  Semantics are presented in Table 34. |
| } |  |  |  |  |
| … |  |  |  |  |

*And add the following text*

The semantics of a *physics* object at node level, are defined in Table 8.2- 2

**Table 8.2- 2: Semantic of a node level physics object**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| needPreciseCollisionDetection | boolean | O | false | If true, the physics engine should handle the collision detection more accurately by increasing the detection rate for this node. |
| linearDamping | number | O | 0 | A non-negative value, in second-1 (s-1), as defined in the international unit system. It defines the linear drag coefficient which corresponds to the rate of decrease of the linear velocity over time.  It is used to compute a new velocity value V(t) at each simulation step (dt):  V(t+dt) = V(t)\*(1-linearDamping\*dt), the velocity being clamped to 0. |
| angularDamping | number | O | 0 | A non-negative value, in second-1 (s-1), as defined in the international unit system. It defines the angular drag coefficient which corresponds to the rate of decrease of the angular velocity over time.  It is used to compute a new velocity value V(t) at each simulation step (dt):  V(t+dt) = V(t)\*(1-angularDamping\*dt), the velocity being clamped to 0. |
| useGravity | boolean | M |  | Indicates if the gravity affects the object |
| mass | number | M |  | Mass of the object in kilogram, as defined in the international unit system. |
| restitution | number | M |  | Provides the ratio of the final to initial relative velocity between two objects after they collide |
| staticFriction | number | M |  | Unitless static friction coefficient as defined in the Coulomb friction model. Friction is the quantity which prevents surfaces from sliding off each other. Static friction is used when the object is lying still. It will prevent the object from starting to move. |
| dynamicFriction | number | M |  | Unitless static friction coefficient as defined in the Coulomb friction model. When a large enough force is applied to the object, a dynamic friction is used and will attempt to slow down the object while in contact with another. |
| } |  |  |  |  |

*In clause 8.2.3 change*

If the scene description document contains a description of physics properties based on another physics model, then that physics model shall take precedence in the processing of the scene.

Otherwise, the application shall handle a physics simulation if the usePhysics Boolean is TRUE on any of the collision trigger extensions defined at the node level. When a collision occurs between two nodes, the application should calculate the combination of the restitution, static friction and dynamic friction values based on the values provided by the collision trigger extension of the two nodes.

*To*

The application shall handle a physics simulation if a physics object is defined at a scene level or/and at any of the nodes. When a collision occurs between two nodes, the application should calculate the combination of the restitution, static friction and dynamic friction values based on the values provided by the physics objects of the two nodes.

*Add the following to Annex A*

A.10 JSON schema for MPEG\_node\_mapping

MPEG\_node\_mapping schema is downloadable from <https://standards.iso.org/iso-iec/23090/-14/amd1/MPEG_node_mapping.schema.json>.

*Add the following sub-section to Annex G*

## G.4 Support for MPEG-I Immersive audio

### G.4.1 General

MPEG-I Immersive Audio has been specified in ISO/IEC 23090-4. The specification assumes the presence of an MPEG-I immersive audio renderer that will receive the MPEG-I audio bitstream, a set of MPEG-H audio streams, as well as information about some scene metadata, such as listener’s pose. It will then use the audio scene metadata in the MPEG-I audio bitstream, the decoded MPEG-H bitstreams, and the pose information to render the spatial audio.

The support of MPEG-I Immersive Audio is achieved by referencing an MPEG-I audio stream in in a MPEG-I scene description document.

The MPEG-I Audio bitstream contains a description of the audio scene that is independent of the main scene description consumed by the Presentation Engine. An alignment between the Presentation Engine and the Audio Renderer is needed, that goes beyond the traditional time alignment but includes also spatial alignment. For that, a mapping need to be established between the node in a MPEG-I scene description document and the node of the audio scene.

The following figure depicts an example of a mapping between a node that contains a car and an external audio node in an MPEG-I Audio bitstream, with a simplified geometry of that car and the attached audio sources. This mapping is described in a The MPEG node mapping extension.

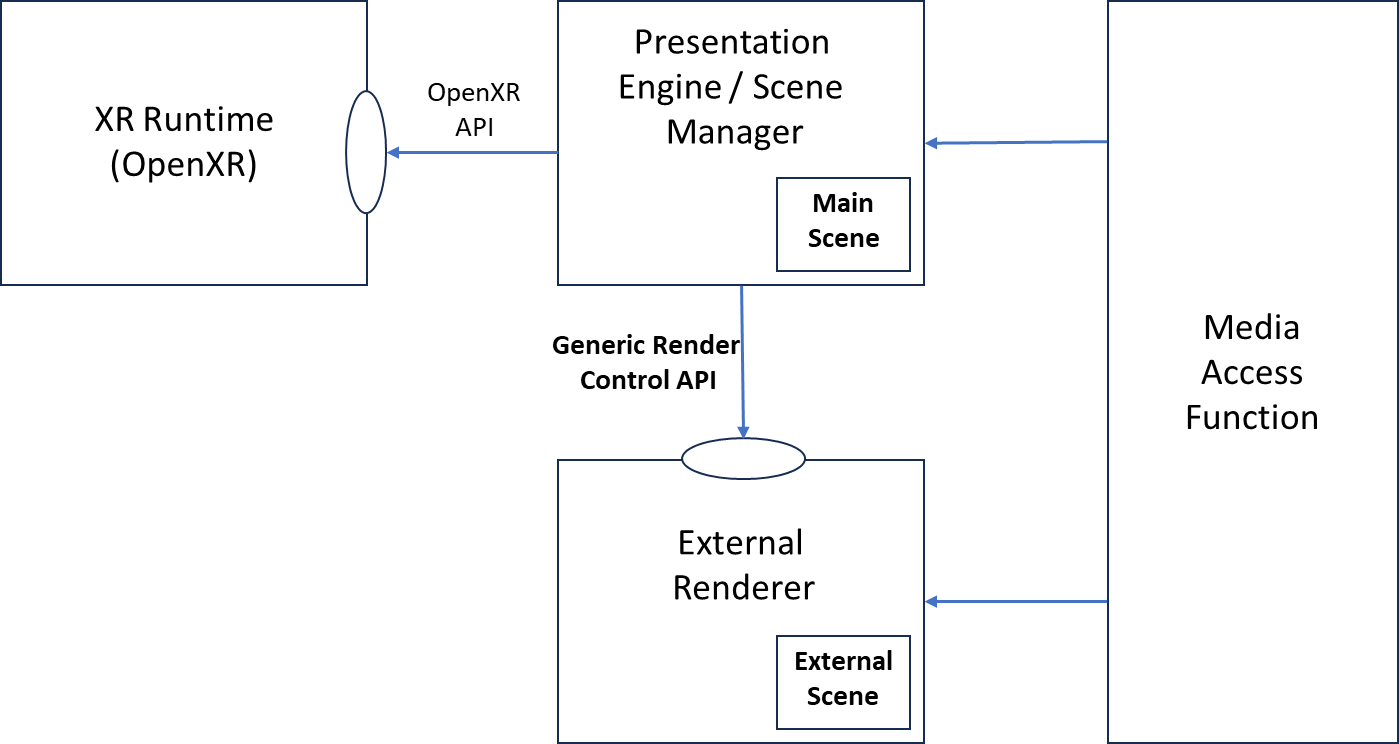
Figure G.4- 1: MPEG-I audio mapping example



This MPEG node mapping extension, identified by MPEG\_node\_mapping, can be used in a broader scope: It establishes a mapping between the node in a MPEG-I scene description document and an external entity, e.g. an MPEG-I audio renderer, that handles a dedicated scene graph, separate from the main scene description. When present, the MPEG\_node\_mapping extension shall be included in a node object.

The architecture for the support of such an external renderer is depicted in the following figure:

Figure G.4- 2: Architecture for external render support



The Generic Render Control API is an abstract API that is offered by external renderers to enable applications, such as Presentation Engines, to control the rendering process by aligning and synchronizing their rendering state to that of the Presentation Engine. This API is used by the Presentation Engine to configure and update the status of the external renderer.

### G.4.2 Generic Render Control API

[editor’s note: Add the API specifications]

### G.4.3 Semantics

The semantics of the MPEG\_node\_mapping extension is provided in the following table. When present, the MPEG\_node\_mapping extension shall be included in a node object.

**Table G.4- 1: Semantic of the MPEG\_node\_mapping extension**

| **Name** | **Type** | **Default** | **Usage** | **Description** |
| --- | --- | --- | --- | --- |
| mappings | array(object) |  | M | An array of mappings associated with the containing node. |
| component | string | “urn:mpeg:sd:component:default” | O | An identifier of the component associated with this mapping. The component may for instance be  “urn:mpeg:sd:component:audio-renderer” to indicate that the component is an audio renderer. |
| source | number | N/A | M | The index in the MPEG\_media that provides the media resource that contains the mapped element. |
| referenceId | number | N/A | M | An identifier of the element in the referenced resource. |
| transform | array(number) | Identity | O | A 4x4 TRS matrix which transforms the 3D coordinates of the node having this glTF extension expressed in the glTF2.0 scene coordinate system to the 3D coordinates of the node of the external renderer graph referenced by the referenceId identifier expressed in the external renderer scene coordinate system.  If the mapped node is a child of an AR Anchor/Trackable, the 4x4 TRS matrix transforms the 3D coordinates of the node having this glTF extension expressed in the AR Anchor/Trackable coordinate system to the 3D coordinates of the node of the external renderer graph referenced by the referenceId identifier expressed in the AR Anchor/Trackable coordinate system. |
| updateRecommendation | object | N/A | O | Indicate update recommendations for the node. The semantics is given in table x2. |
| supportsInteractivity | boolean | false | O | Indicates if interactivity actions applied to the node should be exposed if an API is made available to the Presentation Engine by the renderer of the resource. |

Table G.4- 2: Semantics of an updateRecommendation object

| **Name** | **Type** | **Default** | **Usage** | **Description** |
| --- | --- | --- | --- | --- |
| synchronizationOccurrences | array(enumeration) | [EVENT] | O | An array of synchronization occurrences. Each element of this array is an Enumerator with the following possible values:   * ONCE: the synchronization is done once at the configuration step, * EVENT: the synchronization is done based on the activation of one or more triggers (e.g., visibility, proximity, collision, user input) as those defined in the MPEG\_interactivity extension. The indices of the triggers, from the triggers array of the MPEG\_scene\_interactivity extension, are given in the *events* parameter, * N\_FRAME: the synchronization is periodic every N (1, 2, …) rendering frames. The N value is provided in the *frameNumber* parameter. |
| events | array(number) | N/A | 0 | Array of indices of triggers from the triggers array of the MPEG\_scene\_interactivity extension. Required when EVENT is mentioned in the *synchronizationOccurrences* array. |
| frameNumber | number | N/A | 0 | Indicate the periodicity, in number of frames, of the synchronisation, when N\_FRAME is mentioned in the *synchronizationOccurrences* array. |
| synchronizationOccurrenceCombination | string | “|” | O | A set of logical operations to apply to the synchronization occurrences. A ‘#’ indicates the occurrence index, ‘&’ indicates a logical AND operation, ‘|’ a logical OR operation and ‘~’ a NOT operation. Parenthesis are used to group some operations. Such a syntax may give the following string: “#1&~#2|(#3&#4)”. |

### G.4.4 Processing model

When processing the MPEG\_node\_mapping extension, the Presentation Engine shall identify nodes in the scene description that have a node mapping. The Presentation Engine shall determine if the component identified by the indicated component parameter supports the Generic Render Control API. If it does, the Presentation Engine shall pass the mapping information to the identified component.

The Presentation Engine shall then use the API to align the rendering with the component as configured over the API.

*Add the following Annex I*

Annex I   
(normative)  
  
Support for multi-users interactivity

## I.1 Introduction

This annex is about multi-users application where multiple users meet in a shared space. This space may be described with MPEG-SD and may contain interactivity features as specified in the section 8.2.

The triggers, as specified in the section 8.2.2.1, do not fully support interactions that involve multiple users.

For the TRIGGER\_USER\_INPUT trigger, the trigger is evaluated for each user and the actions associated with the trigger may be performed multiple times, for each received user’s input that met the trigger condition.

For a TRIGGER\_USER\_INPUT trigger involving multiple users, the trigger must be evaluated for all the users and the actions associated with the trigger must be performed only once, when all the user’s inputs met the trigger condition (or when at least one depending on the scenario).

This Annex provides a semantics that address multi-users triggers.

## I.2 Syntax

The following table give the semantic of a trigger object, as specified in the table 51 of the section 8.2.2.1, with a new “multiUsers” parameters added in the *extras* field.

Table 51: semantics of a trigger

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| type | enumeration | M |  | One element that defines the type of the trigger. |
| … |  |  |  |  |
| extras | object | O | N/A |  |
| multiUsers | object | O | N/A | Additional parameters to address a TRIGGER\_USER\_INPUT trigger involving multiple users. See Table I.2- 1 or the details of this object. |

Table I.2- 1: semantic of a multiUsers object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| multiCondition | enumeration | O | ALL | If ALL, the trigger’s condition must be met by all the users.  If ONLYN, the trigger’s condition must be met by exactly N users.  If ATLEASTN, the trigger’s condition must be met by at least N users. |
| duration | number | O | 2 | Give a time duration, in second.  After the trigger condition is met by a first user, it specifies a time window during which the *multiCondition* must be met. |
| number | number | O | 1 | Give the number of users when condition is ONLYN or ATLEASTN. |

I.3 Processing model

After receiving a scene description file, the application parses the file and starts collecting information from all the connected users (user input, pose…).

The application then enters the rendering loop which include a scene update phase performed by the scene manager (SM), that processes the scene interactivity by iterating over the behaviors and manage the other elements of the scene description (animation, lighting, physics…).

The iteration over the behaviors is detailed here after, based on the diagram in the Figure I.3- 1.

During the processing of the scene interactivity, for each behavior and for each trigger referenced in a behavior (1):

* (2) The SM checks if the trigger involves multiple users i.e. a *multiUsers* parameter is specified in the triggers description.
* (3 to 5) If the trigger is not multi-users, for each user information that meet the trigger’s condition, the SM executes the actions referenced in the behavior. It then continues the processing with the next behavior (1).
* (6) If the trigger is multi-users, the SM check if the trigger condition is met for at least one user:
  + if yes, (7) the SM checks if the condition specified in the *multiUsers* object is met, i.e. the *multiCondition* has been met during the given *duration*:
    - If yes, the SM executes the actions referenced in the behavior (8).
    - if no, the processing continues with the next behavior (1).
  + if no, the processing continues with the next behavior (1).

Figure I.3- 1: MPEG-SD triggers processing

