3GPP TSG SA WG4 Meeting #130 S4-241862r01

Orlando, US, 18-22 November 2024

**Source: CATT**

**Title: [FS\_AVATAR] pCR on Avatar Representation of Avatar Properties**

**Agenda item: 9.7**

**Document for: Agreement**

**1. Introduction**

In glTF 2.0, the avatar is described as a node that includes some child nodes, the child nodes include the geometrical mesh, material, texture, etc. as described in Figure 1.

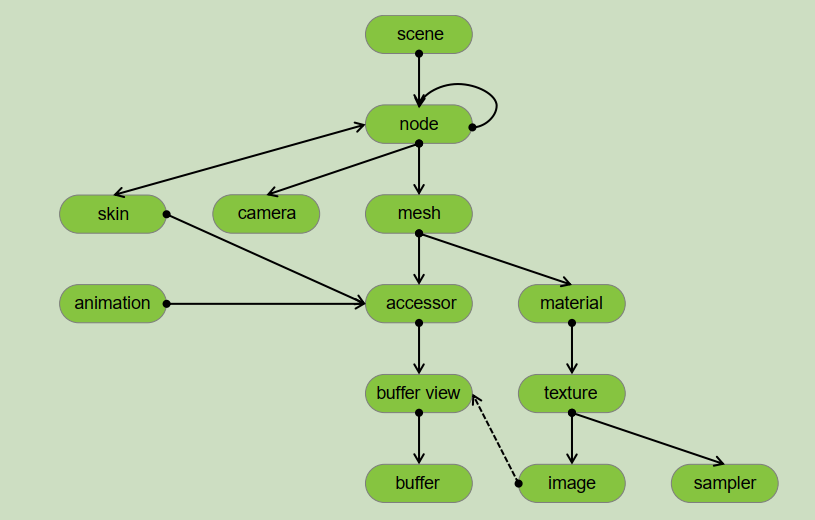


Figure 1. glTF Object Hierarchy in glTF 2.0

In the MPEG-I Scene Description, the avatar is signaled at the node level by the definition of the MPEG\_node\_avatar extension as described in Figure 2. Based on the glTF 2.0 and MPEG-I Scene Description, any node may have child node(s). Each node (including any child nodes of the node) has properties and may have an optional extension property that can be used by this node. The avatar node and its child nodes can reuse all the properties defined by glTF and the extension properties defined by the MPEG-I Scene Description.

*A screenshot of a computer

Description automatically generated*

Figure 2. Object structure with MPEG extensions defined in ISO/IEC 23090-14

As described in clauses 6.3.1, 6.3.23, and 6.3.4, an avatar representation format would include a geometrical model and all associated data (e.g., blend shapes, skeleton, textures, etc.). Some associated data is the data of an avatar property.

The texture is commonly used in avatar representation for some parts of the avatar (e.g., face, head, and/or body). The texture can be mapped from an image as described in glTF 2.0 or from a video stream as described in the MPEG-I Scene Description. During the skinning and animation, if the texture of an avatar part is mapped from a video, the texture is dynamically changed with the video.

**2. Reason for Change**

This document proposes to document the avatar representation for avatar properties.

**3. Proposal**

It is proposed to agree the following changes and to integrate them into 3GPP TR 26.813.

\* \* \* Begin Changes \* \* \* \*

6.3.x Avatar Representation of Avatar Properties

In the MPEG-I Scene Description, the avatar is signaled at the node level by the definition of the MPEG\_node\_avatar extension. Based on the glTF 2.0 and MPEG-I Scene Description, any node may have child node(s). Each node (including any child nodes of the node) has properties and may have an optional extension property that can be used by this node. The avatar node and its child nodes can reuse all the properties defined by glTF and the extension properties defined by the MPEG-I Scene Description.

As described in clauses 6.3.1, 6.3.23, and 6.3.4, an avatar representation format would include a geometrical model and all associated data (e.g., blend shapes, skeleton, textures, etc.). Some associated data is the data of an avatar property.

Some avatar properties can be changed dynamically to create some special effects for the avatar. One example is the texture of the avatar,. the texture is commonly used in avatar representation for some parts of the avatar (e.g., body). The texture can be mapped from an image as described in glTF 2.0 or from a video stream as described in the MPEG-I Scene Description. During the skinning and animation, if the texture of an avatar part is mapped from a video, the texture is dynamically changed with the video.

\* \* \* Next Change \* \* \* \*

7 Reference Architecture

The following figure depicts the reference Architecture for Avatar:



Figure 11. Avatar Reference Architecture

The identified Avatar functions are:

* **Avatar Storage**: an entity that offers storage of base avatars. This entity may be offered by the 5G System, a 3rd party entity, or the local storage of the user’s devices. The Avatar Storage ensures proper access to the base avatar and any related data, including authorization of avatar usage rights. The Authentication functionality should be able to map and identify the ownership of an avatar.
* **Avatar Animation:** depending on the avatar representation format, this entity retrieves the base Avatar, receives representation format-specific animation data streams, and performs the avatar animation to produce the animated avatar that will be used in the rendering process.   
  [Note that some animation approaches may not need to rely on the 3D base avatar, instead they directly produce rendered 2D view of the Avatar.]
* **Scene Management**: creates and composes the shared 3D scene for all participants. It integrates a description of the user’s Avatar and updates its position and orientation based on the user’s pose and updates avatar properties (e.g. the textures if the texture source is changed or the texture is mapped from a video). The updated scene is shared with all participants.
* **Animation data generation:** generating animation data from raw signals. The raw signals may come from cameras, microphones, and specialized motion capturing devices, etc. For example, through the current functional element, the video captured by the camera can be converted into facial feature points, and the audio captured by the microphone can be converted into text, etc.
* **Base Avatar Generation:** generates the base avatar from the inputs such as captured video from camera and other sensors information, possibly in conjunction with a reference avatar. Note that this might be done online or offline.

Figure 12 shows several examples of workflows where avatar functions are differently distributed between the sending UE (UE-A), the network, and the receiving UE (UE-B). The type of avatar data delivered from UE-A (highlighted by the blue arrows) is inherently different depending on which avatar functions are performed on the device by UE-A.



Figure 12. Avatar workflows showing avatar functions performed by different entities

The decision to use a certain avatar workflow depends on the requirements of the specific avatar service, as well as the configurations between the sending (UE-A) and receiving (UE-B) UEs, and the network. More specifically:

* The avatar functions supported by the sending UE (UE-A),
* The avatar functions supported by the receiving UE (UE-B),
* Negotiation of the avatar workflow configuration to request certain avatar functions to be performed by the network.

\* \* \* End of Changes \* \* \* \*