**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2020/M53552**

**April 2020, Alpbach, AT**

|  |  |
| --- | --- |
| **Source** | **Philips, Fraunhofer IIS** |
| **Status** | **Input** |
| **Title** | **Proposed way forward for L3 dynamic updates** |
| **Author** | Jeroen Koppens ([jeroen.koppens@philips.com](mailto:jeroen.koppens@philips.com)), Sam Jelfs ([sam.jelfs@philips.com](mailto:sam.jelfs@philips.com)), Frank Wefers ([Frank.Wefers@iis.fraunhofer.de)](mailto:Frank.Wefers@iis.fraunhofer.de)), Jürgen Herre ([juergen.herre@iis.fraunhofer.de](mailto:juergen.herre@iis.fraunhofer.de)) |

# Introduction

This document presents a compromise between Fraunhofer IIS and Philips for L3 dynamic updates in the MPEG-I Audio CfP subjective tests. An alternative interface was agreed upon, and that a limitation on the number of level 3 interactions per scene is needed in the CfP subjective tests.

# Background

## Motivation

The question if and how dynamic updates shall be realized in the MPEG-I Audio CfP caused a deadlock in the Audio subgroup for some time now. Apart from disagreement on whether L3 scene updates should be evaluated in a subjective test for the CfP, specifically the subtopic of settling on a control interface was a big point of contention.

During the past AhG period Philips and FhG have been working bilaterally towards a compromise. Provided that certain issues were resolved, Philips was willing to support evaluation of level 3 dynamic updates in the CfP subjective tests, in addition to the evaluation based upon technical description that already was supported.

This document develops the alternative approach sketched out in [1] further. It is described how dynamic updates are (pre-) defined in the EIF, how parameters are passed to the AEP and relayed to the Max external renderers. The aim of this input document is providing a solid basis for further discussions solving the deadlock.

## L3 scene update interfaces in the AEP

Below, an overview of dynamic scene update interfaces is given that have been put forward in this discussion. First the interface, jointly conceived in a breakout at MPEG128 autumn 2019 in Geneva, that sparked significant discussion, followed by two other interfaces introduced as a compromise or to illustrate some of the issues with the above interface.

**Contentious interface**

L2: triggerUpdate(updateID)

L3: setSourcePose(srcID, x, y, z, yaw, pitch, roll)

setSourceGain(srcID, gain)

Three particular examples of issues identified with this interface, that are solved by the alternative interfaces are:

* Loss of atomicity of scene updates.
* No explicit restrictions to modifications of audio elements, encoders have to assume all audio elements are potentially modified.
* The need to support string IDs throughout the codec.

**Alternative interface 1 (proposed by Dolby in AhG-call on 14.02.2020)**

L2: triggerUpdate(updateID)

L3: setSourcePose(x, y, z, yaw, pitch, roll, gain)

Inherently this interface is only suitable for the CfP because it allows modification of position, orientation and gain for only one scene element. This resolves several of the issues with the first interface.

Atomicity is guaranteed by the fact that only a single source can be modified and the position and gain are combined in one call. It further solves the incompatibility issue because there is only one place for the limited amount of data to be mapped to.

**Alternative interface 2 (sketched by Philips in input doc m52367 [1])**

L2: triggerUpdate(updateID)

L3: triggerUpdate(updateID, p1, p2, p3, ...)

As an extension of L2 scene updates, this interface allows to define L2 updates with placeholders for actual values, which are later provided when the update is triggered. The EIF describes how parameters p1, p2, etc. map to attributes of scene elements.

Advantages of the interface 2:

* Updates are explicitly pre-defined and in line with L1 and L2 updates.
* L3 updates can now also trigger sound effects, like L2 updates.
* Updates are atomic.
* Encoder knows which elements can be subject to L3 updates, and which parameters of an element are potentially modified by L3 updates.

# Proposal

## Interface

It is proposed to proceed with alternative interface 2.

## Scene restrictions for CfP

For the CfP, it is agreed to maintain the restriction of modifiable attributes in L3 scene updates to position, orientation and gain of audio elements (ObjectSource, ChannelSource, HOASource). These are attributes that all proponents will have available as such in their bitstreams, without risk for incompatibility. I.e. other attributes are more likely to be represented in a different way than in the EIF.

Evaluation of level 3 interactions require specific actions from subjects, and is therefore more complex because a subject must try out all level 3 interactions in a scene to evaluate it thoroughly. Furthermore, it depends to a large extent on what exactly the subject does, how many movements he/she tries to test the interaction. To keep scene evaluation complexity low for the CfP, it is proposed to set a maximum on the number of level 3 scene updates per scene (number tbd), and to strongly encourage scene providers to use the lowest scene update level possible for their scene updates.

## Support by the EIF

The new interface requires dynamic updates to be defined in the EIF, similar to L2 updates. For this purpose, the existing Update and Modify nodes are used. The only difference between an L2 and L3 update is, that the values for the modified attributes are unknown. Instead, these values can be replaced by *parameter indicators*, denoted as $0, $1, etc. (similar to input arguments of a shell script). The following example illustrates this.

<ObjectSource id=”mobileRadio” position=”1 0.6 0.9”

orientation=”0 0 -30”

signal=”sig:radio”

directivity=”dir:radio” />

<Update id=”moveRadio” index=”0” >

<!-- Dynamically update the source’s position and orientation -->

<Modify id=”mobileRadio” position=”**$0 $1 $2**”

orientation=”**$3 $4 $5**”/>

</Update>

<Update id=”radioVolume” index=”1” >

<!-- Dynamically update the radio source’s gain -->

<Modify id=”mobileRadio” gainDb=”**$0**”/>

</Update>

The following conventions are met:

* L1 updates are identified by the existence of the time attribute
* L3 updates distinguish themselves from L2 updates by the occurrence of at least one parameter variable, beginning with ‘$’.
* Parameter variable $0, $1, … may only occur within attribute values of the entity to be modified (not the attributes of the Modify itself).
* Only position, orientation and gainDb attributes may be modified with the parameters.
* It is not possible to pass the entities to be modified (id attribute in Modify) as a parameter.
* The entity to be dynamically modified (i.e. with parameters) must be an Audio Element (ObjectSource, HOASource, ChannelSource)

## Support by the AEP

As only positions, orientations, and gains are to be modified using L3 updates, the type of parameters can be limited to floating point numbers (doubles). This leads to the following revised C-function in the Max externals:

void renderer\_triggerUpdate(t\_renderer\* self, long id, long argc, t\_atom \*argv)

The OSC message from Unity to Max expands the current extOSCAudioTrigger to send both the interaction event ID as well as any parameters required by the update.

This way, the AEP will handle the L3 updates the same as L2 updates, including triggering of sound effect playback if defined in the EIF[[1]](#footnote-1). The only difference is that along with the index, the parameters are provided by Unity and passed on to all Max Externals, using the extended triggerUpdate function.

Example:

When the position/orientation of the radio source shall be dynamically changed, the triggering function is called as follows:

// Change the position/orientation of the radio

triggerUpdate(0, 6, [1.0, -2.01, 2.0, -12, 3, 0]);

# Conclusions

An alternative dynamic scene update interface is proposed for the AEP, restrictions for the CfP test scenes and with minimal required changes to EIF and AEP.

In support of this, FhG has committed to translate the animations of their ParkingLot and DowntownBus scenes to L1 scene updates (static animations), or L2 scene updates in case it is deemed useful to have the user reset the animation.

The authors hope that the Audio subgroup can support this hard-fought compromise.

# References

1. m52367, *Thoughts on scene updates*, MPEG 129

1. A sound effect triggered by an <Update> in the EIF will end up in the TCF and cause the platform to play the sound effect when a certain update index is detected in an interaction event message from Unity. [↑](#footnote-ref-1)