Fraunhofer IIS

Fraunhofer Institute for Integrated Circuits IIS

BBC, Dolby, Fraunhofer IIS

W3C NGA (Next Generation Audio) API Proposal

Fraunhofer IIS

Fraunhofer Institute for Integrated Circuits IIS

Motivation and Introduction to NGA codecs

Bernd Czelhan/23.09.2024

NGA delivers audio experiences that are more accessible, personalized, and interactive – independent of how they are consumed, be it on headphones, soundbars or multi-speaker setups. NGA frees producers from the need to create multiple audio mixes for different reproduction systems by allowing them to deliver a single, multi-purpose audio master instead.«

EBU Technology Factsheet – Next Generation Audio (NGA) ,
https://tech.ebu.ch/docs/factsheets/ebu_tech_fs_nga.pdf

01

Motivation and Introduction

Motivation and Introduction

Example of a native Application – Experience NGA yourself

Eurovision Song Contest 2019 - NGA demonstration - MPEG-H Audio (youtube.com)

Motivation and Introduction

Differences to conventional Audio codecs

Object based audio

The main difference to traditional codecs (AAC, AC-3, MP3, …) is that NGA codecs like (AC-4, DTS-UHD, MPEG-H Audio, ..) do not only transmit encoded audio assets for a fixed channel configuration, but also include a variety of metadata to enable a more suitable user experience

This metadata can be used to enable (incomplete list):

- the user to select a preferred content "version" (authored by the artist/content creator),
- rendering of dynamic objects (audio sources with a time-varying position),
- the system to adapt to the user's loudspeaker setup (including metadata necessary to render dynamic objects),
- the user to adjust the gain/prominence of content components (the content creator is in full control),
- the user to adjust the position of content components (the content creator is in full control),
- ….

We see a need for a dedicated W3C interface to control this advanced functionality to allow a Web experience like the one delivered for native apps.

02

Use-Cases

Cover image: © image copyright

Use Cases

First API requirement

Preselection

A (NGA) Preselection is a pre-defined set of audio components and interactivity values created by the content author (as seen in the video). A good example would be a soccer match with four Preselections:

- "Default" - containing a neutral (all) ambience and a neutral commentator,
- "Home Team" - containing the ambience recorded from the Home Team fan block and a commentator in favor of the Home Team
- "Away Team" - containing the ambience recorded from the Away Team fan block and a commentator in favor of the Away Team
- "Venue" – containing only the neutral ambience

Preselections serve also as entry point for further customization options (if available)

Use Cases

Second API requirement

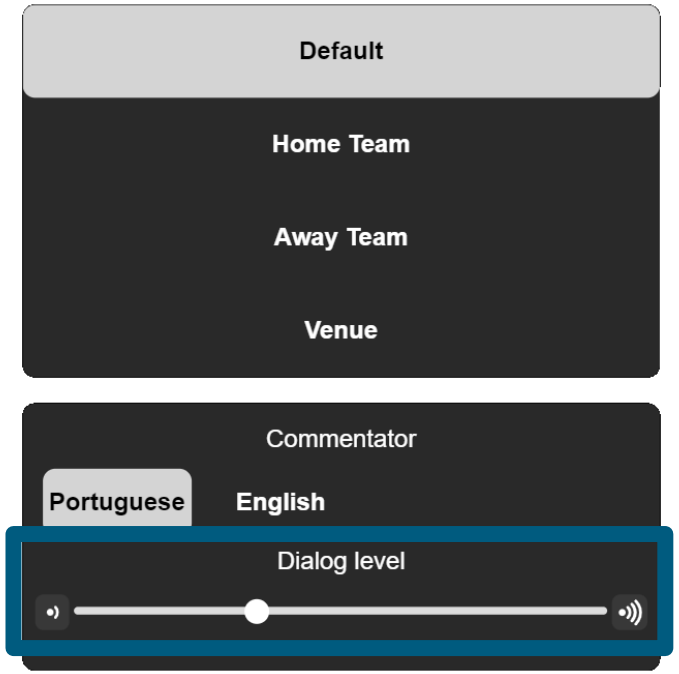
Gain Interactivity

An author of NGA content might allow users to change the gain/prominence of certain content components.

This is normally done for components containing a “dialogue” or “audio description”. Since the preferred dialogue gain/prominence depends on several attributes, e.g. :

- the current listening environment,
- whether the listener is a native or non-native speaker,
- the pronunciation of the speaker,
- the hearing capabilities of the listener,
- ….

it is quite hard to find a “one-value-fits-all” setting. Thus, it is preferable to allow the listeners to customize their experience.



Use Cases

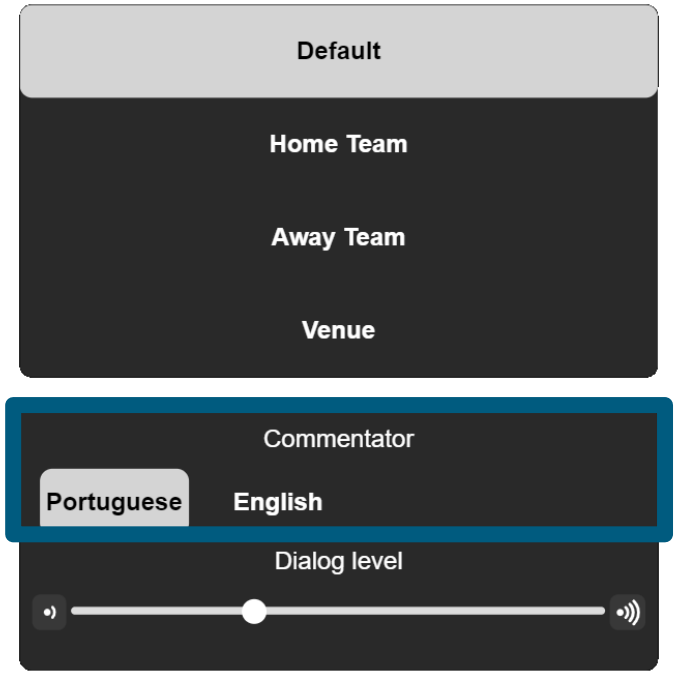
Third API requirement

Selection among multiple audio elements

An NGA bitstream might contain different dialogue languages or content components, which represent alternatives.

The most common example is content components representing different content languages.

Additionally, alternative commentators are quite common.



Use Cases

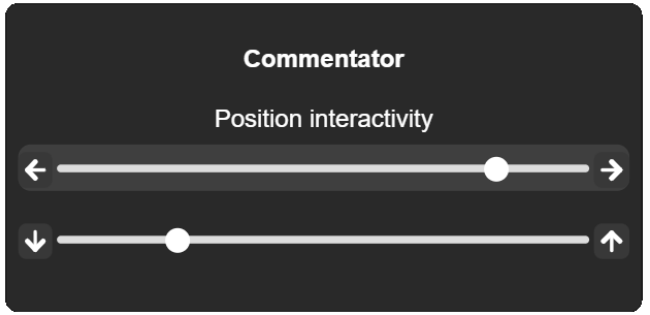
Fourth API requirement

Position Interactivity

A content author of an NGA bitstream might allow users to change the position of certain content components.

This is normally done to increase the spatial separation of components. A possible use case is to move the “audio description” next to the visually impaired user.

Furthermore, it helps to move the “audio description” to the upper level to clearly separate it from the normal audio scene (only possible if there is an upper layer or binaural rendering available).



Use Cases

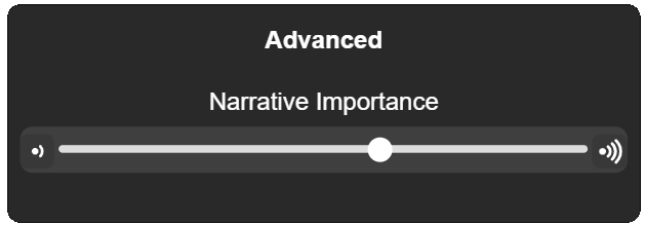
High-level API requirement

Controlling multiple components at the same time

An NGA bitstream might contain different content components, which are important to understand the content author's intent. So, for example not only the dialogue is important for following the plot, but also other audio components (a gunshot, the braking noise of a car, the noise of an approaching wild animal, and so on).

For a good listening experience, it makes sense to group those components into "important" and "less important" components and allow the user to adjust the group of components.

The example shows that this can be done with a single slider, adjusting it to the right will increase the gain of the important components and decrease the background components (and vice versa).



03

API

API
Example NGA Interface

Example NGA Interface

We would like to propose a first version of a possible API extending, for example, HTMLMediaElement.

While designing the proposal, our main goals were:

- the API should cover the full potential of NGA (all main use cases),
- the API should be codec agnostic,
- the API should be in line with the FDIS of ISO/IEC 14496-12, 8th Edition.

We would be happy to get your feedback regarding the draft proposal and we are aware that this is not yet fully defined/streamlined.

We are looking forward to discuss the API with you.

```
function getPreselectionByLabel(ngaAPI, preselectionLabel) {
    const preselections = ngaAPI.preselections;

    for (let i = 0; i < preselections.length; ++i) {
      // search for a specific preselection
      for (let j = 0; j < preselections[i].labels.length; ++j) {
        for (const [key, value] in preselections[i].labels[j].label) {
          if (value === preselectionLabel) {
            return ngaAPI.preselections[i];
          }
        }
      }
    }
    return null;
  }

  const video = document.getElementById("video");
```

Fraunhofer IIS

Fraunhofer Institute for Integrated Circuits IIS

Thank you for your time