



UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

VITOR CAETANO DA SILVA

**Desenvolvimentos de funcionalidades e interfaces para produção
sonora 3D no sistema de áudio espacial AUDIENCE**

São Paulo, SP
2022

VITOR CAETANO DA SILVA

**Desenvolvimentos de funcionalidades e interfaces para produção
sonora 3D no sistema de áudio espacial AUDIENCE**

Relatório parcial apresentado à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, como parte dos requisitos exigidos na disciplina ACH 2018 – Projeto Supervisionado ou de Graduação II, para obtenção do título de Bacharelado em Sistemas de Informação.

Modalidade: TCC longo (1 ano) – individual

Orientador: Prof. Dr. Regis Rossi Alves Faria

São Paulo, SP

2022

Agradecimentos

Agradeço primeiramente ao que tem me ensinado a dar cada passo em rumo à fortaleza. Que nos momentos de angústias me deu conforto e paz. Que nos momentos que eram impossíveis, houve resposta em meu favor. Nunca mais deixarei de estar certo de que sou contigo e enquanto eu viver, amarei o presente que me foi entregue.

Agradeço aos meus pais pelo apoio e educação que me deram durante todo este tempo para contribuir com minha formação como pessoa.

Agradeço aos professores que tiveram a paciência de ensinar os alunos mais de uma vez quando era necessário.

Agradeço ao professor orientador que me auxiliou neste projeto e me ensinou muito sobre computação sônica.

Agradeço a cada aluno desta faculdade que contribuiu para o meu desenvolvimento.

Agradeço a todas as oportunidades de contribuir com o conhecimento.

Resumo

SILVA, Vitor Caetano. **Desenvolvimentos de funcionalidades e interfaces para produção sonora 3D no sistema de áudio espacial AUDIENCE**. 2022. XX f. Monografia (Bacharelado em Sistemas de Informação) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2022.

AUDIENCE é uma biblioteca de objetos e patches para Pure Data (Pd), um ambiente interativo de programação que permite criar e executar funções sonoras em tempo real, ideal para criação e manipulação de efeitos sonoros e prototipagem de funções avançadas para produção e consumo de áudio 3D/360/imersivo. A proposta deste projeto tem como objetivo implementar novas funções e interfaces gráficas de usuário para módulos ainda não integrados ao sistema de áudio espacial AUDIENCE.

O AUDIENCE permite a simulação de cenas espaciais sonoras, ideal para produtores audiovisuais. As novas implementações permitirão integrar novos módulos funcionais, finalizar uma versão de interface de operação nova, e distribuir o pacote em um novo portal construído no Github.

Palavras Chaves: áudio imersivo, cena sonora espacial, computação sônica, produção audiovisual imersiva, ambiente de sonorização espacial

Abstract

SILVA, Victor Caetano. **Development of features and interfaces for 3D sound production in the AUDIENCE spatial audio system**. 2022. XX f. Monograph (Bachelor of Information Systems) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2022.

AUDIENCE is a library of objects and patches for Pure Data (Pd), an interactive programming environment that allows you to create and execute sound functions in real time, ideal for creating and manipulating sound effects and prototyping advanced functions for audio production and consumption. 3D/360/immersive. The purpose of this project is to implement new functions and graphical user interfaces for modules not yet integrated into the AUDIENCE spatial audio system.

AUDIENCE allows the simulation of sound space scenes, ideal for audiovisual producers. The new implementations will allow you to integrate new functional modules, finalize a new op version, and distribute the package in a new portal built on Github.

Keywords: immersive audio, spatial sound scene, sonic computing, immersive audiovisual production, spatial sound environment

Lista de Figuras

Figura 1 - A cadeia de principais funções para auralização utilizada no sistema AUDIENCE

Figura 2 - Arquitetura de camadas funcionais do AUDIENCE, mostrando possíveis sinais entre camadas e modos de saída de áudio

Figura 3 - Fluxograma para criação de show no AUDIENCE

Figura 4 - Módulo S3P Versão 2020

Figura 5 - Print da planilha online no google drive, relativo ao check-list de distribuição do AUDIENCE 2022

Figura 6 - Descreve o relacionamento entre as matrizes de codificação e decodificação.

Figura 7 - Áudio Personalizado, diagrama de exemplo (ITU/EBU 2076)

Figura 8 - Descrição de um objeto AudioStreamFormat em XML

Lista de Tabelas

Tabela 1 - Principais objetos/abstrações disponíveis no AUDIENCE4 Pd versão 2.0.3 e sua função

Tabela 2 - Arquivos de Help

Tabela 3 - Revisão do teste de cada componentes da distribuição

Lista de Abreviaturas e Siglas

PD	<i>Pure Data</i>
WFS	<i>Wave Field Synthesis</i>
S3P	<i>Spatial Sound Scene Production</i>
ITU	<i>International Telecommunication Union</i>
EBU	<i>European Broadcasting Union</i>
BS	<i>Broadcast Service</i>
ADM	<i>Audio Definition Model</i>
NGA	<i>Next Generation Audio</i>
XML	<i>Extensible Markup Language</i>

Sumário

1. Introdução	8
1.1 O Pure Data	8
1.2 AUDIENCE	8
1.3 Objetivos	12
1.4 Materiais e Métodos	13
1.5 Cronograma do Projeto	14
2. Revisão dos componentes do AUDIENCE	15
2.1 Revisão da documentação do "help" dos objetos	17
2.2 Revisão da estrutura e nomenclatura dos componentes existentes na distribuição AUDIENCE	17
3. Metodologia	19
3.1 Correções nas mensagens de erro dos métodos do intra-código	19
3.2 Implantação da nova interface e correção de bugs	19
3.3 Revisão e finalização da Interface Gráfica	19
3.4 Estrutura da nova distribuição disponibilizada no github	19
3.5 Check-list da nova distribuição	20
4. Desenvolvimento da Distribuição do AUDIENCE	21
5. Funcionalidades descritas nas normas da International Telecommunication Union (ITU) / European Broadcasting Union (EBU)	25
5.1 Descrição das Normas ITU/EBU das Recomendações BS.2076 & BS2125 & BS.2051	25
5.2 Biblioteca para a geração dos arquivos	28
6. Conclusão	30
7. Projetos Futuros	31
Anexo A - ITU/EBU2076 - Descrição do Diagrama UML do modelo .ADM	32
Anexo B - ITU/EBU2076 - Tabelas de ATRIBUTOS de cada objeto do arquivo .ADM	33
Anexo C - ITU/EBU2076 - Tabela de SUB-ELEMENTOS dos objetos que compõem o Arquivo .ADM	36
Referências Bibliográficas	42

1. Introdução

1.1 O Pure Data

Pure Data (ou abreviadamente Pd) é uma linguagem de programação visual baseada em fluxo de dados/sinal desenvolvida por Miller Puckette, professor no departamento de música da Universidade da Califórnia, na década de 1990 para processamento de áudio, vídeo e gráficos 2D/3D. Apesar de Miller Puckette ser o seu autor principal, o Pd é uma linguagem opensource com uma base de desenvolvedores bastante extensa que continuam a trabalhar em novas funcionalidades.

No Pure Data você consegue fazer manipulações e processamentos em sinais de áudio construindo um “patch”, que é uma composição de “objetos” conectados. Cada objeto pode receber um número/mensagem/sinal e realiza um processamento nessa entrada recebida e pode “jogar” o dado processado na entrada de outro objeto. Também ainda há vários tipos de botões e visualizadores.

1.2 AUDIENCE

Na área de aplicações de tecnologia na computação sonora e musical, novas demandas tecnológicas surgem para a produção e apresentação de áudio espacial mais realista, de alta resolução, imersiva e interativa.

- Capacidade de localização de sons de forma individual
- Capacidade de manipular um som e recriar seu posicionamento
- Destacar instrumentos específicos no ambiente de escuta
- Adequar a cena acústica produzida ao espaço físico e a acústica local
- Sintetizar um mundo sônico imersivo e interativo
- Criações de ilusões sonoras

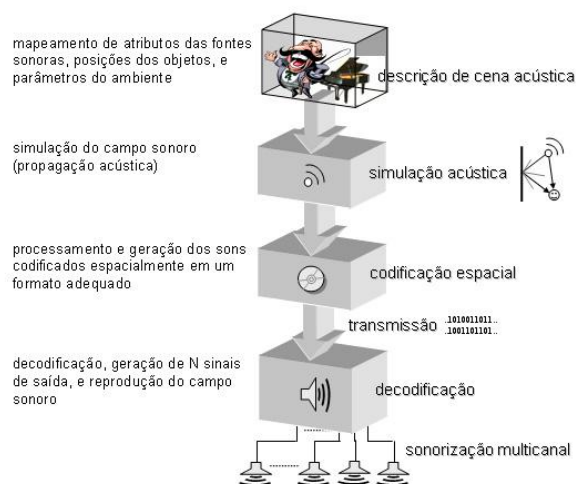
Apresentamos a seguir o sistema de áudio espacial em que trabalharemos, seu estado atual de implementação, e os objetos de desenvolvimento que são desejados no plano de sua nova versão, focalizados nesta proposta.

O projeto AUDIENCE teve início em 2003, tendo como objetivo a pesquisa de áudio envolvente (imersivo ou surround) flexíveis e escaláveis para aplicações de realidade virtual completa. Faria (2005) propôs uma arquitetura modular para auralização de quatro grupos de funções:

- Camada 1: Produção da Cena (Descrição)
- Camada 2: Simulação (Renderização)
- Camada 3: Codificação Sonora Espacial (Descrição temporal e espacial da cena para transmissão)
- Camada 4: Decodificação e Reprodução (Sonorização Multicanal)

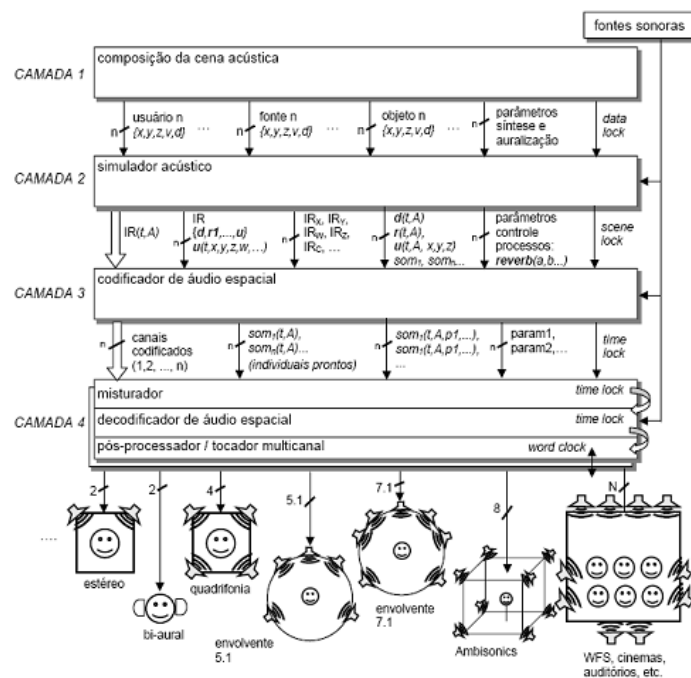
Esta cadeia de ações, dividida por camadas funcionais, é ilustrada na figura a seguir.

Figura 1 - A cadeia de principais funções para auralização utilizada no sistema AUDIENCE



A abordagem proposta na arquitetura AUDIENCE de 4 camadas é única entre os sistemas de produção e reprodução de áudio espacial existentes, sendo amplamente reconfigurável e escalável. A figura abaixo mostra o fluxo de sinais e informações entre as camadas funcionais e os possíveis formatos de apresentação do áudio final por meio de alto-falantes ou fones de ouvido (display sonoro de saída).

Figura 2 - Arquitetura de camadas funcionais do AUDIENCE, mostrando possíveis sinais entre camadas e modos de saída de áudio

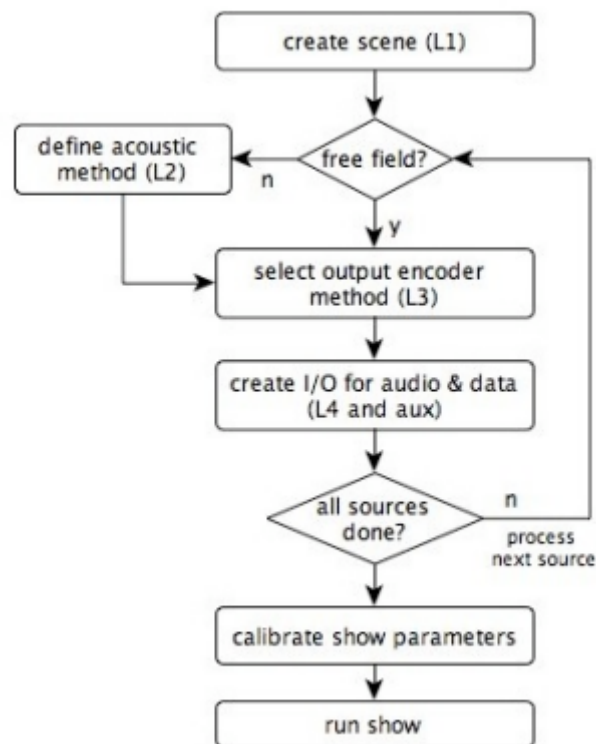


Nesta arquitetura o projetista de um sistema de auralização poderá utilizar soluções que desejar em cada camada, selecionando componentes de diferentes desenvolvedores, e ampliando as possibilidades de compor um sistema que combine as técnicas desejadas.

Nesta atual fase o projeto AUDIENCE amplia seu espectro de ação, aplicando a arquitetura AUDIENCE de sistemas de áudio para uma classe mais ampla de aplicações e problemas em engenharia de áudio, para produzir um maior realismo sonoro nas aplicações puramente sonoras ou nas aplicações audiovisuais, como na TV e cinema digital.

O sistema existe implementado em software e o núcleo básico da distribuição está acessível em <http://ccsl.ime.usp.br/neac/pt-br/openaudience> na sua versão v.1.0.3, release 30/03/12. Há versões posteriores do software mas em estado experimental, contendo módulos não integrados ao programa principal, como um módulo que implementa um novo sistema de produção de cenas sonoras espaciais voltado para produção de trilhas sonoras em 3D/360 graus/imersivo, e um módulo para auralização usando o sistema de Wave Field Synthesis (WFS), que existe numa implementação com um *line-array* para até 20 alto-falantes.

Figura 3 - Fluxograma para criação de show no AUDIENCE



O fluxograma da figura 4 resume as tarefas básicas:

- criação de uma cena sonora (um objeto de L1)
- a configuração de um sistema de renderização acústico (um objeto de L2)
- a codificação do som da cena renderizada (um objeto de L3)
- a decodificação (para um modo de saída específico) e reprodução da cena sonora através de uma plataforma de alto-falantes (um objeto de L4)

Funções auxiliares, como os transportes, controles de mistura e volume estão disponíveis a partir de objetos na camada auxiliar.

1.3 Objetivos

Para esta distribuição do AUDIENCE foi necessário revisar toda a distribuição para que fosse possível organizá-la, e as implementações feitas foram: o desenvolvimento e integração de novas funcionalidades, interfaces, e publicação de uma nova versão do software de áudio espacial AUDIENCE.

As definições das nomenclaturas utilizadas no AUDIENCE sofreram alteração na sua padronização de acordo com as seguintes regras definidas pelo orientador. Houve a necessidade de substituição de todas as nomenclaturas dos componentes do AUDIENCE onde existir as recorrências:

“oa” → “audce” (implementada na primeira fase do projeto)

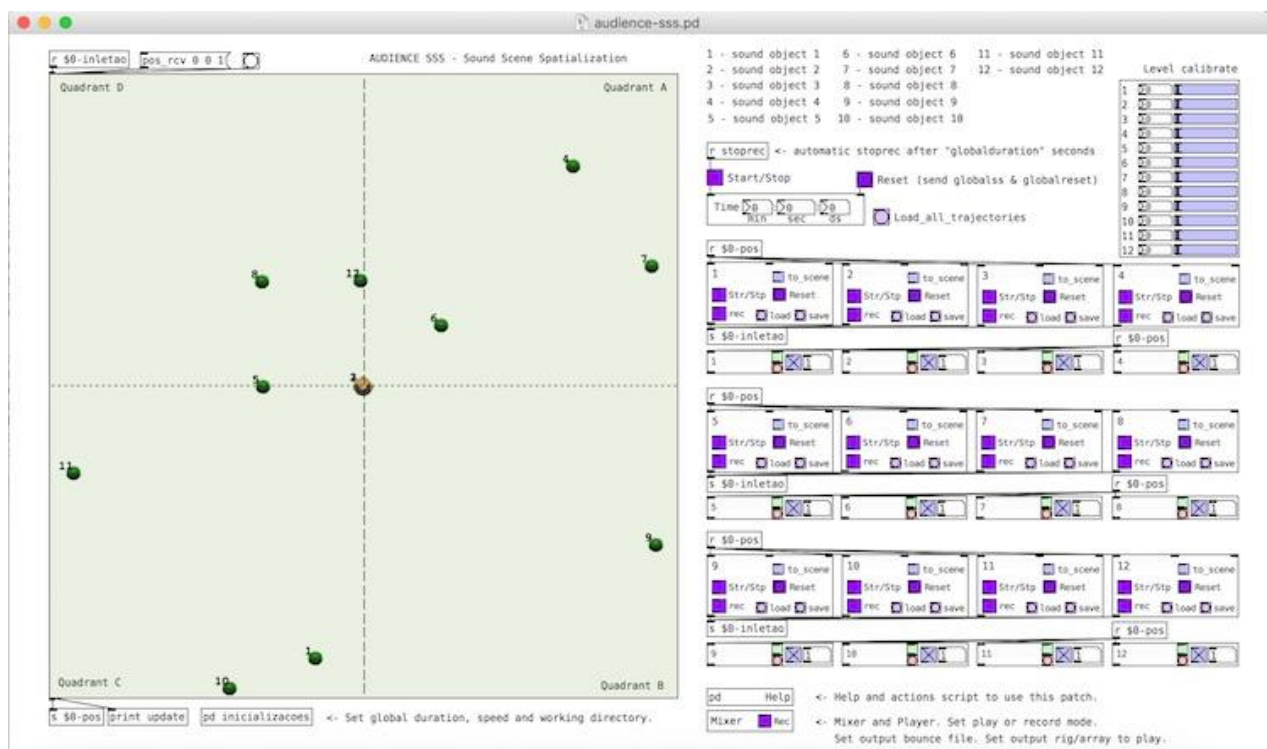
“l1”, “l2”, “l3”, “l4” → “L1”, “L2”, “L3”, “L4”

substituir o “_” → “-” nos nomes dos componentes

nomenclatura do componente “interface-total.pd” → “audce-main.pd”

Após esta etapa, os componentes foram recompilados para o projeto com as novas nomenclaturas, bem como os métodos de criação dos objetos.

O módulo existe atualmente implementado em Pd (vide figura 4, a seguir), utiliza componentes do software AUDIENCE (como objetos gráficos de cena, codificadores e decodificadores Ambisonics). *Figura 4 - Módulo S3P Versão 2020*



A Integração ao sistema AUDIENCE de funcionalidades e compatibilidade com metadados descritores previstos nas normas internacionais da ITU/EBU¹ foi estudada a partir da bibliografia deste trabalho a fim de obter a transposição de arquivos gerados pelo módulo S3P para o formato .ADM.

Os conceitos funcionais e operacionais do Modelo de Definição de Áudio (ADM - *Audio Definition Model*) previsto na norma ITU-R BS.2076-2 e que especifica como os metadados XML podem ser gerados para fornecer as definições de faixas em um arquivo de áudio estão descritos na seção *Desenvolvimento*.

A Integração de funcionalidades e compatibilidade com metadados das normas ITU/EBU foi avaliada e postergada para os projetos que sucederão esta distribuição. A validação dos objetos construídos e de sua documentação (objetos de *help*) foi realizada a fim de se atualizar a distribuição disponível na plataforma do git.

1.4 Materiais e Métodos

Com o auxílio da documentação da norma ITU/EBU que descreve a importação de arquivos .ADM, e de bibliotecas de referência livres, como a Libadm (NIXON ET AL, 2022), serão modelados e então desenvolvidos os códigos necessários para compatibilizar o sistema AUDIENCE com os metadados *XML (eXtensible Markup Language)*, particularmente para permitir entrada/saída de sinalização de tipo de programa multicanal, conteúdo e formas previstas para renderização.

As cenas em formato .ADM são documentadas neste projeto para realizar a importação deste formato de arquivo para o sistema AUDIENCE. A documentação de planejamento é de suma importância para a implementação do código do projeto com o êxito de sua implementação.

¹ A ITU (International Telecommunication Union)/EBU (European Broadcast Union) é a entidade que padroniza as normas que descrevem a criação dos arquivos ADM. Elas mostram as definições que foram padronizadas e como o projeto deve ser utilizado para desenvolver utilizando-se a *libadm* que é a biblioteca disponibilizada online no github.

1.5 Cronograma do Projeto

Primeira Fase do Projeto

- Familiarização com o Sistema AUDIENCE e a plataforma Pure Data
- Revisão da distribuição atual
- Apresentação de Andamento
- Acomodação dos componentes à nova estrutura
- Finalização da Interface Gráfica
- Documentação e primeira publicação da distribuição revisada
- Estudo da Interface do Módulo S3P

Segunda Fase Do Projeto

- Revisão das novas nomenclaturas dos componentes
- Planejamento da documentação para integração da funcionalidade de importação de arquivos ADM previstas nas normas ITU/EBU - BS2076 e BS2125
- Realização de casos de Testes para verificar o funcionamento da importação de arquivos .ADM
- Documentação e preparação da nova distribuição do AUDIENCE
- Atualização dos Helps
- Publicação da nova distribuição do AUDIENCE no github

2. Revisão dos componentes do AUDIENCE

Esta seção descreve a revisão dos componentes da distribuição.

Primeiramente foi feito um estudo sobre o projeto AUDIENCE, este está descrito nesta seção, diretamente descrevendo sua estrutura e organização.

→ Para construir um objeto funcional é necessário a:

- ◆ Definição da sua camada destino: `audce+_camada+_função`
- ◆ Definição do projeto de suas interfaces e processo central: as funções que executa irão utilizar IO
- ◆ Conformidade com o conjunto de mensagens da camada: utilizar os parâmetros definidos utilizados ou endereçados por outros blocos no patch.

O OpenAUDIENCE (OA) foi uma distribuição livre da biblioteca de software, baseado na versão completa. A principal diferença entre eles é a existência de objetos restritos (proprietários) como bibliotecas e códigos-fonte incluídos apenas na versão completa. A distribuição OA exclui todos os componentes restritos (diretórios `/include`, `/src`, bibliotecas e binários proprietários).

A estrutura da biblioteca completa atual está organizada em 4 diretórios:

- **audce**: objetos e abstrações das camadas L1, L2, L3, L4, camada auxiliar, imagens usadas em interfaces, sons de teste, binários extras e bibliotecas de terceiros (necessários em algumas aplicações)
- **app**: diretórios de aplicações e demonstrações
- **src**: arquivos fontes
- **tst**: patches de testes

No **diretório /tst** encontram-se patches de testes úteis para verificar o status de muitas funcionalidades. Por exemplo, o patch `test_audience_volume_patches.pd` resume todas as abstrações de volume disponíveis na versão atual, e o patch `test_audce_L4_objplay~.pd` testa o funcionamento do tocador `audce_L4_objplay~`.

Tabela 1 - Principais objetos/abstrações disponíveis no AUDIENCE4 Pd versão 2.0.3 e sua função

CAMADA	OBJETO	DESCRIÇÃO
L1	audce_L1_gui	GUI de cena para posicionamento de fontes sonoras
L2	audce_L2_allen	Simulador acústico (método fonte-imagem) para uma fonte, c/ saída de Respostas Impulsivas em B-Format (B-IR)
L2	audce_L2_RMgen	Gerador de Matriz de Renderização para n objetos sonoros com saída separada para cada um
L3	audce_L3_amb_3rd~	Codificador Ambisonics até 3a ordem
L3	audce_L3_aacenc	Codificador MPEG-4 AAC
L3	audce_L3_5.1render	Gerador de programa de áudio em modo 5.1 usando metadados produzidos por um gerador de Matriz de Renderização em L2
L3	audce_L3_ambirconv~	Operador de convolução de conjunto de Respostas Impulsivas Ambisonics (B-IR) p/ uma fonte sonora, produzindo sinais em B-Format
L4	audce_L4_amb_3rd~	Decodificador Ambisonics até 3a ordem
L4	audce_L4_aacdec	Decodificador MPEG-4 AAC
L4	audce_L4_amb_rot~	Objeto para rotacionar cena codificada em Ambisonics (campo B-Format) ao redor dos eixos de referência
L4	audce_L4_objplay~	Um tocador de arquivo de fonte sonora, interface c/ file open + transporte
L4	audce_L4_objplay1~	Um tocador de arquivo de fonte sonora, interface c/ file open + transporte + contador de tempo
L4	audce_L4_objplay2~	Um tocador de arquivo de fonte sonora, interface c/ file open + transporte + contador de tempo + controle de volume
AUX	audce_aux_convolver~	Objeto para convoluir fluxo de um sinal de áudio c/ Resposta Impulsiva em tempo real
AUX	audce_aux_counter	Abstração c/ contador de tempo em minutos e parciais
AUX	audce_aux_dbvol~	Pequeno patch para controle de volume em dB
AUX	audce_aux_mixer~	Misturador para n sinais (canais) c/ configuração de nível individual via mensagem
AUX	audce_aux_recordmc~	Abstração GUI para gravar um arquivo sonoro multicanal c/ configuração de no. de canais (até 6) e taxa de amostragem
AUX	audce_aux_playmcfile~	Abstração GUI para reproduzir um arquivo sonoro multicanal c/ configuração de no. de canais (até 6) e taxa de amostragem
AUX	audce_aux_move	Abstração experimental para automatizar o movimento de um objeto sonoro em cena
AUX	audce_aux_objtime	Patch para controlar o tempo de um objeto sonoro em cena no patch principal
AUX	audce_aux_vol~	Patch para controlar o volume of de uma linha de áudio
AUX	audce_aux_objvol~	Patch para controlar o volume de uma linha de áudio com medidor VU
S3P	audce_s3p	Funcionamento do s3p

2.1 Revisão da documentação do "help" dos objetos

Os arquivos de "help" contêm informações sobre os argumentos de criação, os dados retidos pelo objeto (por exemplo, a posição de uma fonte sonora no espaço, a ordem do codificador Ambisonics, etc.), exemplos de uso, os sinais de interface de entrada/saída (ou seja, os sinais válidos que são aceitos nos inlets e sinais disponíveis nos outlets), bugs conhecidos e limitações.

Tabela 2 - Arquivos de Help

Nome	Desc	Revisado ²	Modificações
audce_L1_gui-help	Interface da L1	OK	-
audce_L2_allen-help	Interface da L2	OK	-
audce_L2_RMgen-help	Interface da L2 Rendering Matrix	OK	-
audce_L3_5.1render-help	Gerador de audio 5.1 usando Rendering Matrix	OK	-
audce_L3_amb_3rd~-help	Ambisonics encoder de 3ra ordem	OK	-
audce_L4_amb_3rd~-help	Ambisonics decoder de 3ra ordem	OK	-
audce_L4_objplay~-help	Tocador de som + transporte	OK	-
audce_L4_objplay1~-help	Tocador de som + transporte + tempo	OK	-
audce_L4_objplay2~-help	Tocador de som + transporte + tempo + volume	OK	-
audce_aux_mixer~-help	Mix de sinais de áudio	OK	-
audce_aux_recordmc~-help	Record multichannel	OK	-
audce_aux_time2samples-help	Converte tempo	OK	-
audce_s3p	Módulo s3p	OK	Foi traduzido para o inglês

2.2 Revisão da estrutura e nomenclatura dos componentes existentes na distribuição AUDIENCE

Constata-se que a distribuição é uma só, e que os objetos livres/abertos serão disponibilizados livremente no repositório online, e os objetos alvos de propriedade intelectual só serão acessíveis a usuários que os solicitem e adiram às respectivas licenças dos itens. Tal distinção e restrições de licenças quando houver serão indicados na documentação "help" e intra-código de cada objeto, deixando visível qual licença é aplicável.

² Não foi necessário realizar alterações em todos os helps presentes na distribuição, pois já estavam traduzidos para o inglês e apresentavam as versões com as explicações mais recentes das modificações aplicadas nos projetos.

A nova estrutura de nomeação prevista para a distribuição, é de adotar o termo "audce" como prefixo de todos os objetos e patches do pacote, aplicável aos itens livres/abertos bem como aos itens alvos de propriedade intelectual (IP - Intellectual Property), abandonando o conceito divisório anteriormente adotado de ter duas nomeações coexistentes, uma para a versão livre/aberta ("oa" para OpenAUDIENCE) e outra para a distribuição com objetos não livres por usarem código proprietário de terceiros ou objetos alvo de propriedade intelectual - IP (Intellectual Property).

3. Metodologia

Para desenvolver a nova estrutura do AUDIENCE 2022, são necessárias novas definições para os nomes dos arquivos, correções de mensagens de erro, e implementação nos códigos e na interface.

Exemplo:

Os Arquivos `oa_l1_gui.c`, `oa_l1l4_gui.c`, `oa_l4_gui.c` que faziam parte da antiga distribuição do Open Audience (descontinuada por essa nomenclatura) foram renomeados para `audce_l1_gui.c`, `audce_l1l4_gui.c`, `audce_l4_gui.c` e suas respectivas alterações foram feitas nos métodos do intra-código. Os novos componentes foram organizados para compilar com os novos nomes para serem utilizados na nova distribuição AUDIENCE 2022.

3.1 Correções nas mensagens de erro dos métodos do intra-código

Mensagens de erro do programa com textos que não explicitam ao usuário qual era o erro enfrentado para a criação do componente impossibilita o entendimento sobre qual era o parâmetro que estava sendo enviado de forma errada. Foram revistas para serem atualizadas com frases que explicam melhor qual era o erro enfrentado pelo usuário.

3.2 Implantação da nova interface e correção de bugs

O arquivo da interface gráfica foi testado para se verificar quais eram os bugs existentes, os problemas encontrados estão descritos na seção seguinte.

3.3 Revisão e finalização da Interface Gráfica

A partir da antiga interface-total.pd criada para manter os componentes das camadas do AUDIENCE em uma interface única, um novo visual é estilizado para a ferramenta, utilizando os componentes que já foram desenvolvidos anteriormente.

3.4 Estrutura da nova distribuição disponibilizada no github

O diretório está disponibilizado no github `@latm-lab/audience` disponível em <https://github.com/latm-lab/audience>. Toda sua atualização está disponibilizada junto à documentação.

3.5 Check-list da nova distribuição

O check-list completo da distribuição está compartilhado com o orientador em formato de planilha com todas as abstrações e patches do AUDIENCE junto aos respectivos status de funcionamento e desenvolvimento.

4. Desenvolvimento da Distribuição do AUDIENCE

Primeiramente para a verificação de seus status funcional dos componentes foi utilizado os arquivos de teste disponíveis na pasta “tst” que validam o funcionamento da distribuição AUDIENCE, foi organizado com seus componentes dependentes necessários para o funcionamento e o status que valida o funcionamento atual. Os componentes foram executados em sistema operacional *Windows*, segue visualização em tabela com os resultados das verificações de funcionamento e o componente que o teste descreve:

Tabela 3 - Revisão do teste de cada componentes da distribuição

TIPO	OBJETO DE TESTE (NOME)	COMPONENTES DE TESTE	RESULTADO STATUS ³	Dependências dos componentes
TST	test_5.0_output	ambisonics channels	OK	audce_L1_gui audce_L3_amb_3rd~ audce_L4_amb_3rd~
TST	test_adc+dac	adc stereo and dac multi canal	OK	-
TST	test_ambisonics-coder-decoder	ambisonics encoding e decoding	OK	audce_L1_gui audce_L3_amb_3rd~ audce_L4_amb_3rd~
TST	test_audce_aux_mixer	3 sinais mixados	OK	audce_aux_mixer
TST	test_audce_aux_move	ferramenta de movimento automático	OK	audce_gui_grid audce_L3_amb_3rd~ audce_L4_amb_3rd~
TST	test_audce_aux_recordmc~	auxiliar record mc:c/ n° de canais	OK	audce_L1_gui
TST	test_audce_L1_gui	automação de movimento do receptor na cena e mensagens da L1_gui	OK	audce_L1_gui audce_gui_grid
TST	test_audce_L1_scene_controller	Determina qual cena está ativa enquanto a cena existe especificando a região no espaço parametrizado	OK	audce_L1_gui audce_L1_scene_controller
TST	test_audce_L2_allen	Allen I2 que parametriza a cena, objetos e fontes	Erros	audce_L1_gui audce_L2_allen
TST	test_audce_L4_amb_rot~	audce_L4_amb_rot~ com mensagens de orientação e elevação	Erros	audce_L1_gui audce_L3_amb_3rd~ audce_L4_amb_rot~
TST	test_audce_L4_objplay~	arquivo e transporte do audce_L4_objplay~	OK	-
TST	test_audce_L4_objplay~_2objects	audce_L4_objplay com 2 objetos	OK	-
TST	test_audce_L4_objplay1~	arquivo e transporte do audce_L4_objplay~ com controle de volume	OK	-
TST	test_audce_L4_objplay2~	audce_L4_objplay com 2 objetos e volume e tempo	OK	-
TST	test_audce_mic_delay	testa mic com delay	Erros	audce_aux_convolver

³ Observações: Erros estão associados ao *audce_aux_convolver* o qual ainda não foi possível utilizar a biblioteca fftw3 para compilar sua dll, pois a versão da fftw3 foi atualizada e não dá suporte a versão da época em que foi utilizada no AUDIENCE.

TST	test_audience_install	instalação básica de todo o AUDIENCE	Erros	audce_L2_RMgen audce_L3_aacenc audce_L3_amb_mix~ audce_aux_convolver~ audce_L3_MPSEnc audce_L2_err audce_L4_aacdec audce_L4_MPSEdec
TST	test_audience_volume_patches	patches de volume	OK	-
TST	test_audience-scene	Teste de cena	Erros	audce_L1_gui audce_L3_amb_3rd~ audce_L4_amb_3rd~

Foram encontrados alguns erros para verificação de todos os componentes do AUDIENCE. Estes erros foram relacionados para posteriormente verificá-los em novas distribuições.

Mensagens de Erro:

Outra das necessidades foi verificar as mensagens de erro. As mensagens de erro dos componentes do AUDIENCE foram todas verificadas em seu intra-código. As mensagens de erro do programa, continham textos que não explicitavam ao usuário qual era o erro enfrentado para a criação do componente e não se entendia para o tipo de situação qual era o parâmetro que estava sendo enviado de forma errada.

Para isso foi necessário mudar o entendimento e a sintática das mensagens e adicionar informações mais completas sobre estes erros. A seguir estão as modificações que foram feitas com a orientação do professor orientador para que oferecesse um melhor entendimento ao usuário sobre o erro para criar o componente:

- No arquivo do componente de interface gráfica da camada L4 de nome “audce_l4_gui.c” tivemos a alteração da seguinte recorrência do erro:
 - a linha de erro *post("audce_l4_gui : ERROR. Arguments used are not numeric");* teve as suas recorrências modificada para:
 - *post("audce_l4_gui : ERROR. At least one argument used is not numeric");*
- Da mesma forma, tivemos alterações no componente da interface gráfica de L1L4, que une as camadas L1 e L4 em uma única interface gráfica. As alterações feitas em “audce_l1l4_gui.c” foram:
 - a linha de erro *post("audce_l1l4_gui : ERROR. wrong argument type");* para:
 - *post("audce_l1l4_gui : ERROR. At least one argument used is not numeric");* na primeira ocorrência
 - *post("audce_l1l4_gui : ERROR. Argument 5 must be numeric");* na segunda ocorrência
 - *post("Argument 6 must be a symbol");* na terceira ocorrência

- E também foram necessárias as alterações no intracódigo da interface gráfica do componente da camada L1 “audce_l1_gui.c”:
 - a linha de erro `post("audce_l1_gui : wrong arguments");` teve sua linha modificada para:
 - `post("audce_l1_gui : ERROR. Arguments used are not numeric");` na primeira ocorrência
 - `post("audce_l1_gui : ERROR. At least one argument is not numeric");` na segunda ocorrência
 - a linha de erro `post("audce_l1_gui: pos_src - número de argumentos errado!");` teve sua linha modificada para:
 - `post("audce_l1_gui: pos_rcv - No. of arguments must be 3.");` na primeira ocorrência
 - `post("audce_l1_gui: pos_src - argumentos não numéricos");`
 - `post("audce_l1_gui: pos_rcv - At least one of used arguments is not numeric");` na segunda ocorrência
 - `post("audce_l1_gui: pos_rcv - Fonte inexistente!");`
 - `post("audce_l1_gui: pos_rcv - Source not found");` na terceira ocorrência

Correção de Bugs - Verificação dos limitadores no patch da interface gráfica:

O arquivo da interface chamava-se “*interfacetotal.pd*” e foi atualizado para “main-audce.pd”, além disso teve as seguintes correções feitas:

- Os painéis da interface possuem uma espécie de intra-painel gráfico que seta todas as suas funções e o que será mostrado na interface gráfica principal, eles foram abertos para que pudessem ser alterados e assim foram setados os números máximos para bloquear os inputs evitando assim um valor inválido. Desta forma os valores setados foram valores padrões escolhidos, foi utilizado para este padrão os valores 100 para os metros máximos do ambiente da cena. 20 para o número de objetos da cena e 12 para fontes sonoras presentes na cena, esta solução evitou que valores muito grandes pudessem ser colocados gerando erros e até a paralisação total do programa Pure Data. A documentação do help do s3p foi atualizada para ser utilizada na interface gráfica desta nova distribuição do AUDIENCE. As criações do help para a interface gráfica estão previstas para a próxima etapa.

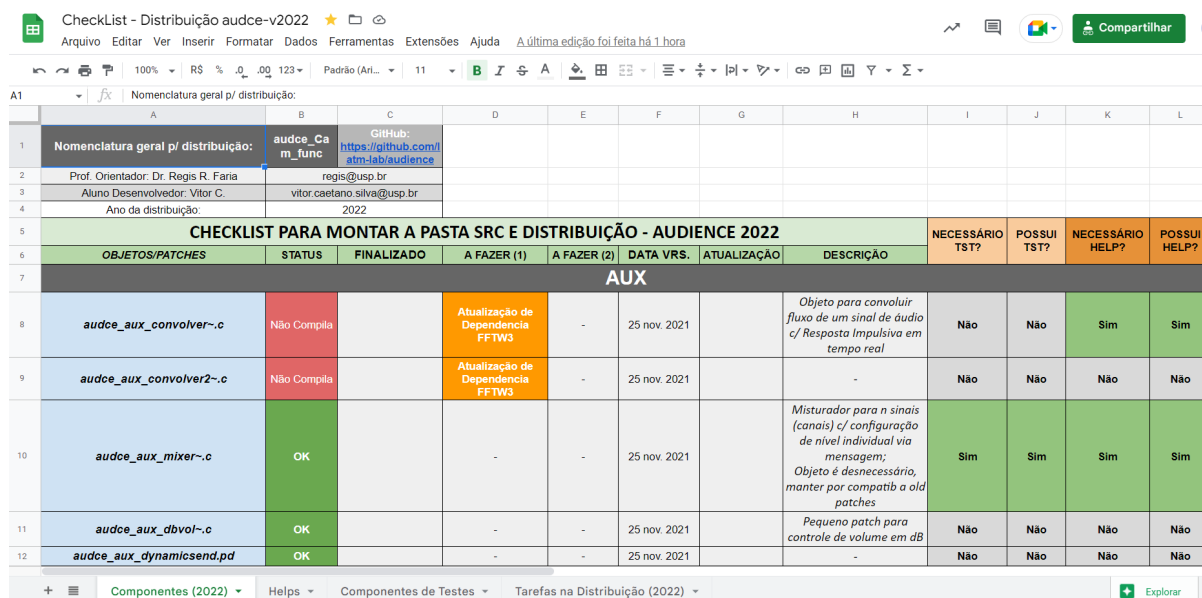
Versionamento:

A versão está disponibilizada no github, e está distribuída nas pastas: “audce_2022” que é a versão da nova distribuição; o “audce_x86_64” que se trata da versão principal que deve ser distribuída e compilada para os SOs Linux, Mac e Windows 32 ou 64 bits; A antiga versão do OpenAudience já não está mais presente no repositório, com a redefinição da nomenclatura.

Check-list:

O checklist faz a organização da distribuição do AUDIENCE, desta forma é possível que o versionamento melhore a partir desta distribuição, de forma a tornar compreensível a visão geral do projeto. Todos os componentes estão organizados e separados pelo seu nível de camada. Os componentes são descritos com as principais informações: **Objetos/Patches, Status, Finalizado, A fazer (1 e 2), Data de Versão, Atualização, Descrição, e informações sobre os arquivos TST e HELP.**

Figura 5 - Print da planilha online no google drive, relativo ao check-list de distribuição do AUDIENCE 2022



CheckList - Distribuição audce-v2022												
Arquivo Editar Ver Inserir Formatar Dados Ferramentas Extensões Ajuda A última edição foi feita há 1 hora												
100% R\$ 123 Padrão (Ari... 11												
Nomenclatura geral p/ distribuição:												
1	Nomenclatura geral p/ distribuição:	audce_Ca m_func	GitHub: https://github.com/atm-lab/audience									
2	Prof. Orientador: Dr. Regis R. Faria	regis@usp.br										
3	Aluno Desenvolvedor: Vitor C.	vitor.caetano.silva@usp.br										
4	Ano da distribuição:	2022										
5	CHECKLIST PARA MONTAR A PASTA SRC E DISTRIBUIÇÃO - AUDIENCE 2022								NECESSÁRIO TST?	POSSUI TST?	NECESSÁRIO HELP?	POSSUI HELP?
6	OBJETOS/PATCHES	STATUS	FINALIZADO	A FAZER (1)	A FAZER (2)	DATA VRS.	ATUALIZAÇÃO	DESCRIÇÃO				
7	AUX											
8	audce_aux_convolver~.c	Não Compila		Atualização de Dependência FFTW3	-	25 nov. 2021		Objeto para convoluir fluxo de um sinal de áudio c/ Resposta Impulsiva em tempo real	Não	Não	Sim	Sim
9	audce_aux_convolver2~.c	Não Compila		Atualização de Dependência FFTW3	-	25 nov. 2021		-	Não	Não	Não	Não
10	audce_aux_mixer~.c	OK		-	-	25 nov. 2021		Misturador para n sinais (canais) c/ configuração de nível individual via mensagem; Objeto é desnecessária, manter por compatib a old patches	Sim	Sim	Sim	Sim
11	audce_aux_dbvol~.c	OK		-	-	25 nov. 2021		Pequena patch para controle de volume em dB	Não	Não	Não	Não
12	audce_aux_dynamicseend.pd	OK		-	-	25 nov. 2021		-	Não	Não	Não	Não
+ Componentes (2022) Helps Componentes de Testes Tarefas na Distribuição (2022) Explorar												

5. Funcionalidades descritas nas normas da International Telecommunication Union (ITU) / European Broadcasting Union (EBU)

A implementação das funcionalidades consistem em inserir ao projeto os desenvolvimentos das normas descritas nos projetos dos manuais ITU/EBU que descrevem os arquivos ADM: BS.2076 e BS2125. Estas servem para definir a leitura de arquivos muito requisitados no ramo de criação de cenas áudio espaciais, os arquivos ADM descrevem a cena sonora de forma padrão para diversas aplicações áudio espaciais.

Suas especificações estão descritas nos documentos que estão disponibilizados online no site da ITU e esta documentação apresenta como a biblioteca deve ser utilizada e quais foram as normas atribuídas para a recomendação. Estas bibliotecas são encontradas no github distribuídas livremente na rede da internet.

Acessando-se o link do projeto de distribuição pública no github de cada um desses projetos, a partir da pesquisa pelos repositórios na internet, podemos utilizar as bibliotecas e instalá-las para poder utilizar suas funções e criar a tradução entre os dados do projeto, de acordo com a especificação documentada nas referências bibliográficas das normas da ITU. A partir da recomendação descrita no manual, as principais atribuições de cada biblioteca, as normas definidas e suas dependências estão descritas nesta seção.

5.1 Descrição das Normas ITU/EBU das Recomendações BS.2076 & BS2125 & BS.2051

No Reino Unido, a BBC tem feito experiências com áudio baseado em objetos, o que levou a uma nova recomendação da ITU (ITU-R BS.2125 “*The representation serial of the Audio Definition Model*”), publicada em fevereiro de 2019.

O áudio para transmissão e cinema está evoluindo para uma experiência imersiva e interativa, que requer o uso de formatos de áudio mais flexíveis. Uma abordagem baseada em canal fixo não é suficiente para abranger esses desenvolvimentos e assim combinações de canal, objeto e baseado em cena formatos estão sendo desenvolvidos.

O requisito central para permitir que todos os diferentes tipos de áudio sejam distribuídos, seja por arquivo ou por streaming, seja qual for o formato de arquivo/stream usado, os metadados devem coexistir para descrever completamente o áudio.

Cada faixa individual dentro de um arquivo ou stream deve poder ser corretamente renderizada, processada ou distribuída de acordo com os metadados que o acompanham.

Para garantir a compatibilidade em todos os sistemas, o Áudio Definition Model é um padrão aberto que tornará isso possível.

O diagrama geral do modelo é dado na Figura X. Isso mostra como os elementos se relacionam entre si e ilustra a divisão entre as partes de conteúdo e formato. Ele também mostra o pedaço de um BS.2088 e como ele conecta as faixas do arquivo ao modelo. Onde um arquivo BS.2088 contém várias trilhas de áudio, é necessário saber o que cada trilha é.

O trecho contém uma lista de números correspondentes a cada faixa no arquivo. Assim, para um Arquivo de 6 faixas, a lista tem pelo menos 6 faixas. Para cada faixa há um número *audioTrackFormatID* e um número *audioTrackUID* (observe o 'U' adicional que significa 'único'). A razão da lista pode ser maior que o número de faixas é que uma única faixa pode ter diferentes definições em momentos diferentes, portanto, exigirá vários *audioTrackUIDs* e referências. O *audioTrackFormatID* é usado para procurar a definição do formato dessa faixa específica. o *audioTrackFormatIDs* não são exclusivos; por exemplo, se um arquivo contiver cinco pares estéreo, haverá cinco *audioTrackFormatIDs* idênticos para descrever o canal 'esquerdo' e cinco para descrever o canal 'direito' canal. Portanto, apenas dois *audioTrackFormatIDs* diferentes precisarão ser definidos. No entanto, *audioTrackUIDs* são únicos (daí o 'U'), e eles estão lá para identificar exclusivamente a faixa. este o uso de IDs significa que as faixas podem ser ordenadas de qualquer maneira no arquivo; seus IDs revelam o que aquelas faixas são.]

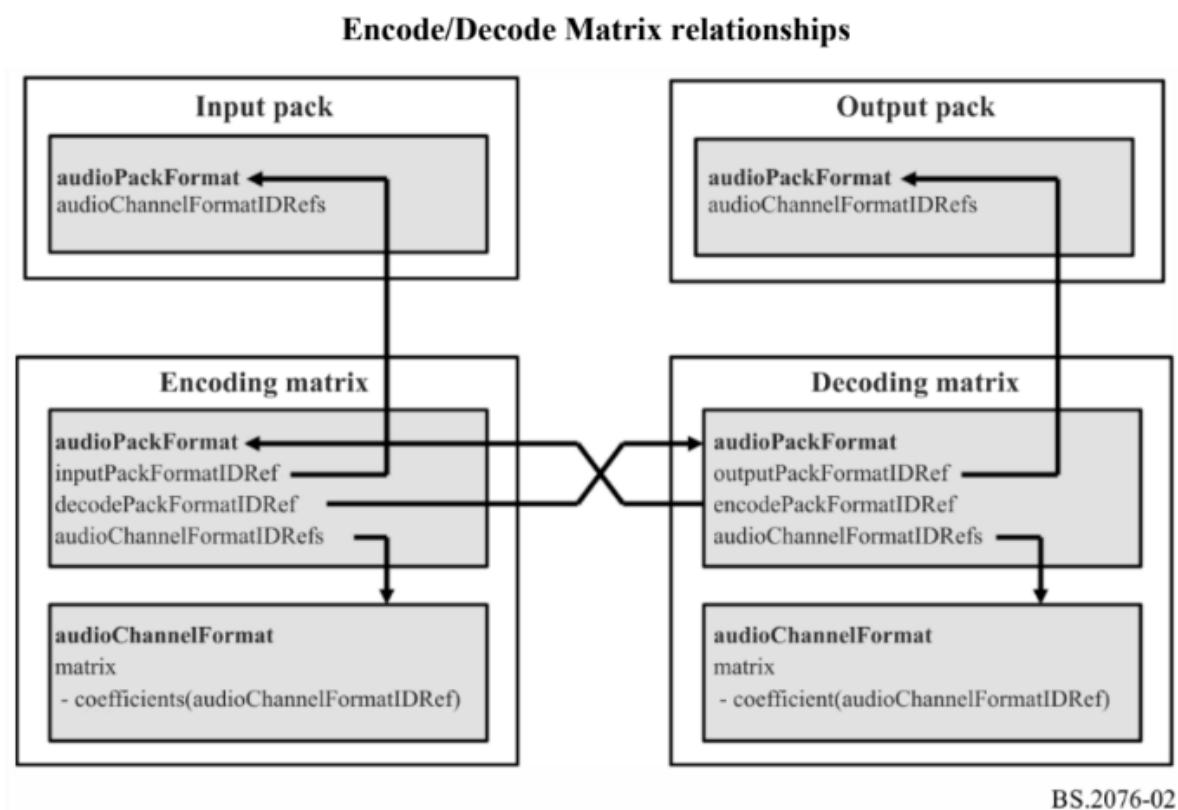
Usando uma cena de áudio com cinco pares estéreo como exemplo, o *audioTrackFormat* define qual faixas são esquerda e direita, não quais pertencem juntas, nem o que é representado nelas.

O **AudioObject** é usado para determinar quais faixas devem estar juntas e onde elas estão no arquivo. Este elemento liga os dados de áudio reais com o formato, e é aí que entra o *audioTrackUID*.

Múltiplos elementos *audioProgramme* podem ser definidos em uma representação de árvore ADM XML. este facilita a descrição de uma apresentação que representa um número predefinido de misturas significativas que os usuários podem escolher. Cada elemento *audioProgramme* pode referenciar apenas um subconjunto de elementos *audioContent* da árvore XML ADM. Este é um método para permitir que o ADM descreva áudio personalizado.

As tabelas e diagramas que descrevem de forma objetiva os atributos utilizados para descrever o arquivo ADM estão disponíveis nos Anexos A, B e C.

Figura 6 - Descreve o relacionamento entre as matrizes de codificação e decodificação.



O arquivo ADM utiliza IDs para fazer a codificação e decodificação dos objetos. Se o typeDefinition do audioPackFormat for definido como Matrix, haverá subelementos extras

disponível para permitir a definição de codificação

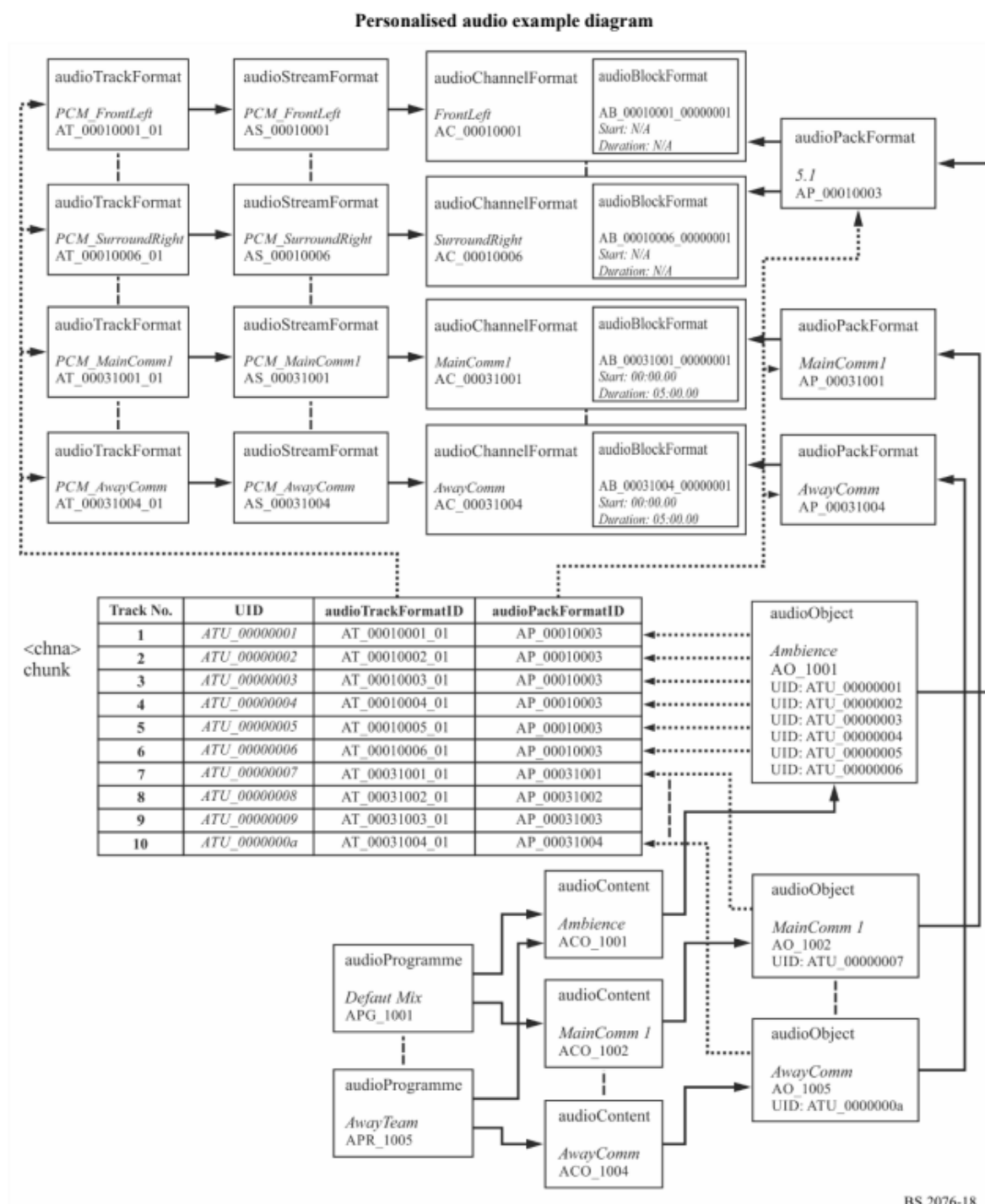
A matriz pode ser uma codificação, uma matriz de decodificação ou uma matriz direta. Uma matriz de codificação converte um audioPackFormat de entrada de qualquer tipo em um audioPackFormat codificado em matriz. uma decodificação matrix pega audioPackFormat codificado em matrix e converte em uma saída baseada em canal audioPackFormat.

O diagrama na Figura anterior mostra como os audioPackFormats da matriz do codificador e do decodificador se relacionam, bem como nos audioPackFormats de entrada e saída e audioChannelFormats.

Pelo diagrama podemos compreender como o arquivo .ADM é estruturado e organizado. Sua comunicação é feita pela biblioteca libadm.

O diagrama a seguir mostra um exemplo de descrição de áudio de uma cena complexa:

Figura 7 - Audio Personalizado, diagrama de exemplo (ITU/EBU 2076)



5.2 Biblioteca para a geração dos arquivos

libadm é a biblioteca que descreve uma especificação para os metadados que podem ser usados para descrever áudio baseado em objeto, áudio baseado em cena e áudio baseado em canal. BS.2127) e permite o monitoramento em qualquer alto-falante ITU-R BS.2051.

Também inclui a documentação para a recomendação BS.2076 que também descreve a estrutura de metadata para leitura de arquivos ADM. Os requisitos necessários para a instalação da biblioteca estão listados abaixo:

- *compiler with C++11 support*
- *Boost header libraries (version 1.57 or later) (<https://www.boost.org/>)*
- *Boost.Optional*
- *Boost.Variant*
- *CMake build system (version 3.5 or later) (<https://www.cmake.org/>)*

A biblioteca é utilizada para ler arquivos que são gerados em estruturas de linguagem XML (Extensible Markup Language) Ela é necessária para implantar a criação dos arquivos .ADM no sistema AUDIENCE. Um exemplo de código é mostrado a seguir exibindo a estrutura descrita nas tabelas de descrição dos objetos que estão no Anexo B.

Figura 8 - Descrição de um objeto AudioStreamFormat em XML

```
<audioStreamFormat audioStreamFormatID="AS_00010001"
audioStreamFormatName="PCM_FrontLeft" formatDefinition="PCM"
formatLabel="0001">
  <audioTrackFormatIDRef>AT_00010001_01</audioTrackFormatIDRef>
  <audioChannelFormatIDRef>AC_00010001</audioChannelFormatIDRef>
</audioStreamFormat>
```

6. Conclusão

O objetivo previsto neste projeto foi a revisão da distribuição do AUDIENCE, para isto foi necessário aplicar técnicas de organização do projeto, testes e validações e estudar os componentes que já haviam sido construídos, bem como revisar documentações.

A nova distribuição AUDIENCE está com correções das mensagens de erro e conta com atualizações dos helps. A nova nomenclatura foi definida como “audce” e a antiga versão Open Audience foi descontinuada.

Na segunda fase do projeto foi feito o estudo para que se possa implementar a leitura de arquivos no formato .ADM que são recorrentes na área de áudio espacial e cinema, que contam com sons de alta qualidade e espacialização 3D.

Neste projeto não houve esta implantação devido à sua grande complexidade de estruturar e adequar ao AUDIENCE. O estudo feito para a implantação permite a realização deste projeto em implementações futuras. A seção a seguir apresenta as próximas etapas que o projeto AUDIENCE irá contemplar.

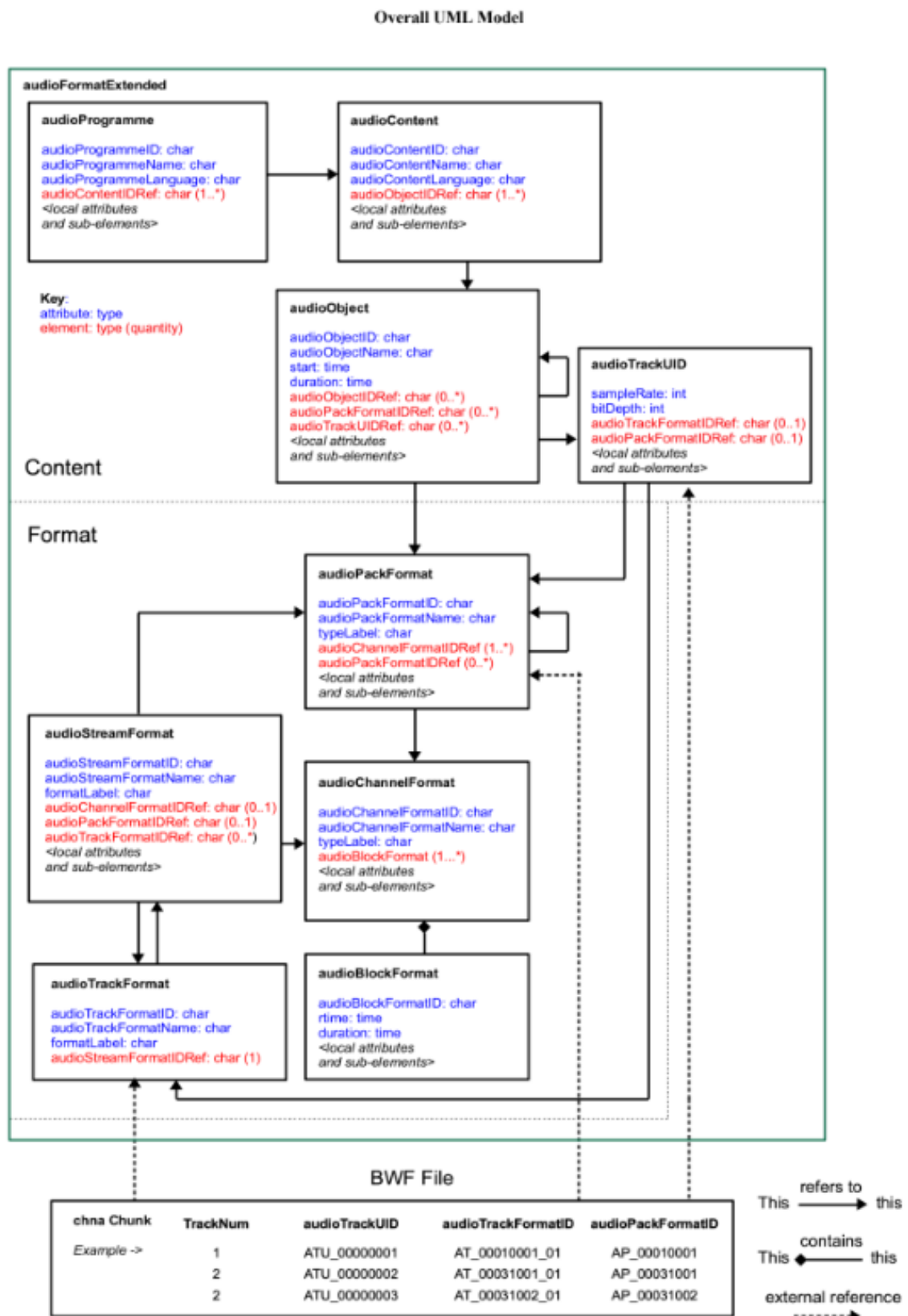
7. Projetos Futuros

Para próximas distribuições do AUDIENCE as seguintes implementações estão listadas abaixo:

- Importação e Exportação do arquivo ADM.
- A implementação do grau de compatibilidade aos metadados e arquivos do tipo *Broadcast Wave 64Bit (BW64) audio file format* previsto na norma ITU-R BS.2088-1 (*Long-form file format for the international exchange of audio programme materials with metadata*)
- Adicionar funcionalidades para sistemas de som avançados da NGA (*Next Generation Audio*) como a renderização para reproduzir em sistemas de alto-falantes flexíveis conforme previsto na norma ITU-R BS.2051-2 (*Advanced sound system for programme production*)⁴;
- Implementar um renderizador 3D conforme previsto na norma ITU-R BS.2127-0 (*Audio Definition Model renderer for advanced sound systems*);

⁴ Um sistema de som avançado considera uma configuração de reprodução além daquelas especificadas na Recomendação ITU-R BS.775 ou um sistema com qualquer configuração de reprodução que possa suportar sinais de entrada baseados em canais, objetos ou cenas ou sua combinação com metadados.

Anexo A - ITU/EBU2076 - Descrição do Diagrama UML do modelo .ADM



Anexo B - ITU/EBU2076 - Tabelas de ATRIBUTOS de cada objeto do arquivo .ADM

Tabelas que descrevem os atributos de cada objeto descrito no arquivo .ADM

AudioTrackFormat attributes

Attribute	Description	Example	Required
audioTrackFormatID	ID for track, see § 6. The yyyy digits of AT_yyyxxxxx_nn represent the type of audio contained in the track. The yyyxxxxx digits should match the audioStreamFormat yyyxxxxx digits	AT_00010001_01	Yes
audioTrackFormatName	Name for track	PCM_FrontLeft	Yes
formatLabel	Descriptor of the format	0001	Optional
formatDefinition	Description of the format	PCM	Optional

audioTrackFormat sub-elements

Sub-element	Description	Example	Quantity
audioStreamFormatIDRef	Reference to an audioStreamFormat	AS_00010001	1 (see Note below)

audioStreamFormat sub-elements

Sub-element	Description	Example	Quantity
audioChannelFormatIDRef	Reference to audioChannelFormat	AC_00010001	0 or 1
audioPackFormatIDRef	Reference to audioPackFormat	AP_00010003	0 or 1
audioTrackFormatIDRef	Reference to audioTrackFormat	AT_00010001_01	0...* (see Note below)

audioChannelFormat attributes

Attribute	Description	Example	Required
audioChannelFormatName	Name of the channel	FrontLeft	Yes
audioChannelFormatID	ID of the channel, see § 6 for the use of the audioChannelFormatID in typical channel configurations. The yyyy digits of AC_yyyxxxxx represent the type of audio contained in the channel. The xxxx digits should match the audioStreamFormat xxxx digits.	AC_00010001	Yes
typeLabel	Descriptor of the type of channel	0001	Optional *
typeDefinition	Description of the type of channel	DirectSpeakers	Optional *

* At least one of typeLabel or typeDefinition is required.

audioBlockFormat attributes

Attribute	Description	Example	Required
audioBlockFormatID	ID for block	AB_00010001_00000001	Yes
rtime	Start time of block (relative to the start time of the parent audioObject) The start time is in the time format as described in § 5.11.	00:00:00.00000 or 00:00:00.00000S48000	Optional Default when not present: 00:00:00.00000
duration	Duration of block. The duration is in the time format as described in § 5.11.	00:00:05.00000 or 00:00:05.00000S48000	Optional Default when not present: Unbounded duration

audioPackFormat attributes

Attribute	Description	Example	Required
audioPackFormatID	ID for the pack, see § 6 for the use of the audioPackFormatID in typical channel configurations. The yyyy digits of AP_yyyxxxxx represent the type of audio contained in the pack.	AP_00010001	Yes
audioPackFormatName	Name for the pack	stereo	Yes
typeLabel	Descriptor of the type of channel	0001	Optional*
typeDefinition	Description of the type of channel	DirectSpeakers	Optional*
importance	Importance of a pack. Allows a renderer to discard a pack below a certain level of importance. 10 is the most important, 0 is the least.	10	Optional

audioProgramme attributes

Attribute	Description	Example	Required
audioProgrammeID	ID of the programme	APR_1001	Yes
audioProgrammeName	Name of the programme		Yes
audioProgrammeLanguage	Language of the dialogue content contained in this programme (as a String). It is recommended to use a language code to identify the language. The language code can be given as a 2- or 3-character code as specified by ISO 639-1 or ISO 639-2. Both ISO 639-2/B and ISO 639-2/T may be used.	fr	Optional
start	Start time for the programme. The start time is in the time format as described in § 5.11.	00:00:10.00000 or 00:00:10.00000S48000	Optional
end	End time for the programme. The end time is in the time format as described in § 5.11.	00:10:00.00000 or 00:10:00.00000S48000	Optional
maxDuckingDepth	Indicates the maximum amount of automatic ducking allowed for every audioObject in the programme. Range is 0 to -62 dB		Optional

audioObject attributes

Attribute	Description	Example	Required	Default
audioObjectID	ID of the object	AO_1001	Yes	-
audioObjectName	Name of the object	dialogue_stereo	Yes	-
start	Start time for the object, relative to the start of the audioProgramme. The start time is in the time format as described in § 5.11.	00:00:00.00000 or 00:00:00.00000S48000	Optional	00:00:00.00000
duration	Duration of object. The duration is in the time format as described in § 5.11.	00:02:00.00000 or 00:02:00.00000S48000	Optional	duration of audioProgramme
dialogue	If the audio is not dialogue set a value of 0; if it contains only dialogue a value of 1; if it contains both then a value of 2.	0	Optional	2
importance	Importance of an object. Allows a renderer to discard an object below a certain level of importance. 10 is most important, 0 least.	10	Optional	10
interact	Set to 1 if a user can interact with the object, 0 if not.	1	Optional	0
disableDucking	Set to 1 to disallow automatic ducking of object, 0 to allow ducking	0	Optional	0

Anexo C - ITU/EBU2076 - Tabela de SUB-ELEMENTOS dos objetos que compõem o Arquivo .ADM

Os sub-elementos estão presentes para descrever a cena, podendo ser obrigatórios ou opcionais.

audioChannelFormat sub-elements

Sub-element	Description	Attributes	Quantity
audioBlockFormat	Time division of channel containing dynamic metadata	See § 5.4	1...*
frequency	Describes the high and/or low cut-off frequency for the audio in Hz	typeDefinition = "lowPass" or "highPass"	0...2

Common audioBlockFormat sub-elements

Sub-element	Attribute	Description	Units	Example	Quantity	Default
gain	gainUnit	Definition of a gain value to be applied to all audio samples corresponding to the audioBlockFormat. An optional gainUnit attribute (either 'linear' or 'dB') can be used to define the unit of the gain value. The default unit is 'linear'. For a detailed description of the application of this gain value see § 12.	gain value, default is linear value	0.5 (linear), -6 (dB)	0 or 1	1.0
importance		Importance of the audioChannelFormat, defined for the duration of the current audioBlockFormat.	0 to 10	10	0 or 1	10
headLocked		Indicates if the perceived location of the audio element is locked to the head (flag = 1) or not locked (flag = 0) See § 9.3	0/1 flag	1	0 or 1	0

Sub-element	Attribute	Description	Units	Example	Quantity	Default
headphoneVirtualise	bypass	Specifies whether the object should be virtualised using a headphone virtualiser or not (1=renderer to stereo, 0=renderer with headphone virtualiser) See § 9.4	1/0 flag	1	0 or 1	0
	DRR	Direct to Reverberant Ratio in dB. See § 9.4	dB	-130..130	0 or 1	130 (anechoic-all direct)

audioBlockFormat sub-elements for DirectSpeakers

Sub-element	Attribute	Bound attribute	Description	Units/ Values	Example	Quantity
speakerLabel		N/A	A reference to the label of the speaker position	–	M-30	0...*
position	coordinate=“azimuth”		Exact azimuth location of sound	Degrees	–30.0	1
position	coordinate=“azimuth”	max	Max. azimuth location of sound	Degrees	–22.5	0 or 1
position	coordinate=“azimuth”	min	Min. azimuth location of sound	Degrees	–30.0	0 or 1
position	coordinate=“elevation”		Exact elevation location of sound	Degrees	0.0	1
position	coordinate=“elevation”	max	Max. elevation location of sound	Degrees	5.0	0 or 1
position	coordinate=“elevation”	min	Min. elevation location of sound	Degrees	0.0	0 or 1
position	coordinate=“distance”		Exact normalized distance from origin	Normalized to 1	1.0	0 or 1
position	coordinate=“distance”	max	Max. normalized distance from origin	Normalized to 1	0.8	0 or 1

audioBlockFormat sub-elements for Matrix

Sub-element	Attribute	Description	Quantity	Default
outputChannelFormatIDRef*	–	For defining a decoding or direct matrix, this is the output audioChannelFormat that defines the channel being decoded to.	0 or 1	
jumpPosition		If jumpPosition is set to 1 the position will change instantly from the previous block’s position. If set to 0 then interpolation of the position will take the entire length of the block.	0 or 1	0
	interpolationLength	If the interpolationLength attribute is used, and the jumpPosition value is 1, then the interpolation will take as long as the specified value. The interpolation length should be shorter or equal than the block’s duration and it should be specified in seconds (at least 5 d.p).	0 or 1	Duration of block
matrix	–	See Table 14.	1	

* This element name has been editorially changed from *outputChannelIDRef*, which was the incorrect name used in the original version of BS.2076-1. Therefore, ADM parsing software should be aware that *outputChannelIDRef* might occur in some files instead of *audioChannelFormatIDRef* and should be able to read both.

matrix sub-elements						
Sub-element	Attribute	Description	Units	Example	Quantity	Default
coefficient	gainUnit	Unit for attribute of 'gain'. If gainUnit is not used, 'linear' unit is assumed.		linear / dB	0 or 1	'linear'
coefficient	gain	Multiplication factor of another channel. Constant value. Type: float	Linear or logarithmic gain value*	−0.5	0...* Note: no more than one use of each attribute can be specified.	1.0
coefficient	gainVar	Multiplication factor of another channel. Variable. Type: string (reference to float)	A variable representing a linear gain value*	clev		-
coefficient	phase	Phase shift of another channel. Constant value. Type: float	degrees	90		0
coefficient	phaseVar	Phase shift of another channel. Variable. Type: string (reference to float)	A variable representing a value in degrees	ph		-
coefficient	delay	Time delay of another channel. Constant value Type: float	ms (float)	10.5		0.0
coefficient	delayVar	Time delay of another channel. Variable Type: string (reference to float)	A variable representing a time in ms	del		-
coefficient		Reference to another audioChannelFormat ID		AC_0001000 1	1...*	

audioBlockFormat sub-elements for Objects (polar)

Sub-element	Attribute	Description	Units	Example	Quantity	Default
position	coordinate= "azimuth"	azimuth "theta" of sound location	degrees (−180 ≤ theta ≤ 180)	−22.5	1	
position	coordinate= "elevation"	elevation "phi" of sound location	degrees (−90 ≤ phi ≤ 90)	5.0	1	
position	coordinate= "distance"	distance "r" from origin, where 1 is on the unit sphere surface	relative distance value	0.9	0 or 1	1.0
width		horizontal extent	degrees (0 to 360)	45	0 or 1	0.0
height		vertical extent	degrees (0 to 360)	20	0 or 1	0.0
depth		distance extent	Ratio (0 to 1)	0.2	0 or 1	0.0

audioBlockFormat sub-elements for Objects (Cartesian)

Sub-element	Attribute	Description	Units	Example	Quantity	Default
position	coordinate="X"	left/right dimension	Relative Units	−0.2	1	
position	coordinate="Y"	back/front dimension	Relative Units	0.1	1	
position	coordinate="Z"	bottom/top dimension	Relative Units	−0.5	0 or 1	0.0
width		X-width	Relative Units (0 to 1)	0.03	0 or 1	0.0
depth		Y-width	Relative Units (0 to 1)	0.05	0 or 1	0.0
height		Z-width	Relative Units (0 to 1)	0.07	0 or 1	0.0

audioBlockFormat sub-elements for Objects

Sub-element	Attribute	Description	Units	Example	Quantity	Default
cartesian		Specifies coordinate system, if the flag is set to 1 the Cartesian coordinate system is used, otherwise spherical coordinates are used.	1/0 flag	1	0 or 1	0
diffuse		Describes the diffuseness of an audioObject (if it is diffuse or direct sound)	0.0 to 1.0	0.5	0 or 1	0
channelLock	maxDistance	If set to 1 a renderer can lock the object to the nearest channel or speaker, rather than normal rendering. The optional maxDistance attribute defines the radius of a sphere around the object's position. If one or more speakers exist in the defined sphere or on its surface, the object snaps to the nearest speaker. If maxDistance is undefined, a default value of infinity is assumed, meaning that the object should snap to the nearest of all speakers (unconditioned channelLock).	1/0 flag for channelLock, float value for maxDistance in the range from 0.0 to 2 ⁽¹⁾ sqrt(3)	1, 1.0	0 or 1	0 (channel Lock), infinity (maxDistance)
objectDivergence	azimuthRange ⁽¹⁾	Adjusts the balance between the object's specified position and two other positions specified by the azimuthRange value (symmetrical on both sides of the object at the object's position +/- azimuthRange). A value of 0 for the objectDivergence means no divergence. This attribute shall only be used when the coordinate system is spherical.	0 to 1.0 for objectDivergence, 0.0 to 180.0 (angle) for azimuthRange	0.5, 60.0	0 or 1	0.0, 0.0
	positionRange ⁽¹⁾	Adjusts the balance between the object's specified position and two other positions specified by the positionRange value (symmetrical on both sides of the object at the object's position +/- positionRange along the X-axis). A value of 0 for the objectDivergence means no divergence. This attribute shall only be used when the coordinate system is Cartesian.	0 to 1.0 for objectDivergence, 0.0 to 1.0 for positionRange	0.5, 0.25	0 or 1	0.0, 0.0

Sub-element	Attribute	Description	Units	Example	Quantity	Default
jumpPosition		If jumpPosition is set to 1 the position will change instantly from the previous block's position. If set to 0 then interpolation of the position will take the entire length of the block. If the interpolationLength attribute is used, and the jumpPosition value is 1, then the interpolation will take as long as the specified value. The interpolation length should be shorter or equal than the block's duration.	1/0 flag	1, 0.05125	0 or 1	0
	interpolationLength	If the interpolationLength attribute is used, and the jumpPosition value is 1, then the interpolation will take as long as the specified value. The interpolation length should be shorter or equal than the block's duration.	seconds (at least 5 d.p)	0.05125	0 or 1	Duration of block
zoneExclusion ("zone" sub-elements)		Indicates which speaker/room zones the object should not be rendered through.	see "zone" sub-elements		0 or 1	
zone (sub-element of zoneExclusion)	minX maxX minY maxY minZ maxZ	Specifies the corner points of a cuboid in the 3D space that will be excluded from rendering for Cartesian coordinates. Multiple zone elements can be used to specify more complex exclusion shapes.	−1.0 to 1.0 float for each Cartesian attribute. String for a label to describe the exclusion zone	minX=−1.0 maxX=1.0 minY=−1.0 maxY=0.0 minZ=−1.0 maxZ=1.0 "Rear half"	1.. ⁽¹⁾	
	minElevation maxElevation minAzimuth maxAzimuth	Specifies the circular projection onto the sphere for spherical coordinates. Multiple zone elements can be used to specify more complex exclusion shapes.	−180 to 180 float for the spherical azimuth attribute and −90 to 90 float for the spherical elevation attribute. String for a label to describe the exclusion zone	maxElevation=30 minElevation=−30 minAzimuth=−30 maxAzimuth=30 "Centre front"	1.. ⁽¹⁾	
screenRef		Indicates whether the object is screen-related (flag is equal to 1) or not (flag is equal to 0)	1/0 flag	0	0 or 1	0

Sub-element	Attribute	Description	Units/Type	Example	Quantity
audioComplementaryObjectGroupLabel	language	Definition of a label for a group of complementary audioObjects. The language attribute can be used for definition of multiple audioComplementaryObjectGroup labels in different languages. See Table 27.	String	“主音声” language=“jp”	0...*
audioComplementaryObjectIDRef		Reference to another audioObject that is complementary to the object, e.g. to describe mutually exclusive languages.	ID String	AO_1003	0...*
audioTrackUIDRef		Reference to an audioTrackUID (when using a BW64 file according to [7] this is listed in the <chma> chunk)	ID String	ATU_0000000 1	0...*
audioObjectInteraction		Specification of possible user interaction with the object.	-	-	0 or 1
gain	gainUnit	Definition of a gain value to be applied to all audio samples referenced by the audioObject. The default value is 1.0. An optional gainUnit attribute (either ‘linear’ or ‘dB’) can be used to define the unit of the gain value. The default unit is ‘linear’. For a detailed description of the application of this gain value see § 12.	Linear or logarithmic gain value	0.5 (linear), -6.0 (dB)	0 or 1
headLocked		Indicates if the perceived location of the audio element is locked to the head (flag = 1) or not locked (flag = 0) See § 9.3 Default Value is 0	0/1 flag	1	0 or 1
positionOffset (when polar coordinates are used)	coordinate=“azimuth”	Apply an offset to the “azimuth” angle to all elements in the audioObjects.	Degrees	30.0	0 or 1
	coordinate=“elevation”	Apply an offset to the “elevation” angle to all elements in the audioObjects.	Degrees	15.0	0 or 1
	coordinate=“distance”	Apply an offset of “distance” to all elements in the audioObjects.	Normalised distance	0.9	0 or 1
positionOffset (when Cartesian coordinates are used)	coordinate=“X”	Apply an offset of “X” axis to all elements in the audioObjects.	Normalised value	-0.2	0 or 1
	coordinate=“Y”	Apply an offset of “Y” axis to all elements in the audioObjects.	Normalised value	0.1	0 or 1
	coordinate=“Z”	Apply an offset of “Z” axis to all elements in the audioObjects.	Normalised value	-0.5	0 or 1
mute		Status of the audioObject to play back or not. Set to 0 if the object is played back (default). Set to 1 if the object is muted.		1	0 or 1
alternativeValueSet	alternativeValueSetID	An alternative set of parameters that will be used if the alternativeValueSetID is referenced by an audioProgramme or audioContent element. See § 5.6.5 for sub-elements.			0...*

audioProgramme sub-elements

Sub-element	Attribute	Description	Example	Quantity
audioProgrammeLabel	language	Definition of audioProgramme label. The language attribute can be used for definition of multiple audioProgramme labels in different languages. The language code should be given as a 2- or 3-character code as specified by ISO 639-1 or ISO 639-2. Both ISO 639-2/B and ISO 639-2/T may be used	“Venue” language=“en”	0...*
audioContentIDRef		Reference to content	ACO_1001	1...*
loudnessMetadata	-	See § 5.8.4		0 ... *
audioProgrammeReferenceScreen	-	Specification of a reference/production/monitoring screen size for the audioProgramme, see § 5.8.3. If the reference screen-size is not given, a default screen-size is implicitly defined (see § 10.6).		0 or 1
authoringInformation		See § 5.8.6		0 or 1
alternativeValueSetIDRef		Reference to an alternativeValueSet within an audioObject.	AVS_1001_0001	0...*

Referências Bibliográficas

MPEG-H. ISO/IEC - *High Efficiency Coding and Media Delivery in Heterogeneous Environments..* Disponível em: <<https://www.mpeg.org/standards/MPEG-H/>> [S.l.] Acesso em: 6 abr. 2022.

SONY. *360 WalkMix Creator. Plugin software compatible with popular digital audio workstations.* Disponível em: <<https://360ra.com/>> Acesso em: 6 abr. 2022.

FRAUNHOFER, Institute for Integrated Circuits IIS. *MPEG-H Audio. Next Generation Audio.* Disponível em: <<https://www.mpeg.org/standards/MPEG-H/>> Acesso em: 6 abr. 2022.

Fórum do Sistema Brasileiro de TV Digital Terrestre. *TV 3.0 Project.* 2020. Disponível em: <https://forumsbtvd.org.br/tv3_0/> Acesso em: 6 abr. 2022.

Faria et al. **Projeto AUDIENCE e OpenAUDIENCE.** Disponível em: <<http://ccsl.ime.usp.br/neac/audience>> Acesso em: 01 de Abril de 2022.

Nixon, Thomas et al. **Libear - An Open-Source Library for the EBU ADM Renderer.** Disponível em: <<https://github.com/ebu/libear>>. Acesso em: 8 abr. 2022.

Nixon, Thomas et al. **Libadm- An Open-Source Library for the EBU ADM Renderer.** Disponível em: <<https://github.com/ebu/libadm>>. Acesso em: 8 abr. 2022.

Nixon, Thomas et al. **Libbw64- An Open-Source Library for the EBU ADM Renderer.** Disponível em: <<https://github.com/ebu/libbw64>>. Acesso em: 8 abr. 2022.

Institut für Elektronische Musik und Akustik - IEM. *Pure Data - Pd Community Site,* Áustria, 2022 Disponível em: <<https://puredata.info/>> Acesso em: 1 abr. 2022.

International Telecommunication Union. 2018, 2019. *Radio Communication Sector of ITU - Recommendation BS.2051-2 ITU-R: BS Series Broadcasting service (sound)*

Electronic Publication Geneva, Suíça, 2018. Disponível em:
<https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.2051-2-201807-I!!PDF-E.pdf>

International Telecommunication Union. 2018, 2019. *Radio Communication Sector of ITU - Recommendation BS.2076-2 ITU-R: BS Series Broadcasting service (sound)*
Electronic Publication Geneva, Suíça, 2019. Disponível em:
<https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.2076-2-201910-I!!PDF-E.pdf>

International Telecommunication Union. 2018, 2019. *Radio Communication Sector of ITU - Recommendation BS.2088-1 ITU-R: BS Series Broadcasting service (sound)*
Electronic Publication Geneva, Suíça, 2019. Disponível em:
<https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.2088-1-201910-I!!PDF-E.pdf>

International Telecommunication Union. 2018, 2019. *Radio Communication Sector of ITU - Recommendation BS.2125-0 ITU-R: BS Series Broadcasting service (sound)*
Electronic Publication Geneva, Suíça, 2019. Disponível em:
<https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.2125-0-201901-I!!PDF-E.pdf>

International Telecommunication Union. 2018, 2019. *Radio Communication Sector of ITU - Recommendation BS.2127-0 ITU-R: BS Series Broadcasting service (sound)*
Electronic Publication Geneva, 2019. Disponível em: <https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.2127-0-201906-I!!PDF-E.pdf>

J. Riedmiller, S. Mehta, N. Tsingos and P. Boon, *Immersive and Personalized Audio: A Practical System for Enabling Interchange, Distribution, and Delivery of Next-Generation Audio Experiences*. ***SMPTE Motion Imaging Journal***, vol. 124, no. 5, pp. 1-23, July/August 2015, doi: 10.5594/j18578.

FARIA, R. R. A. AUDIENCE for Pd, a scene-oriented library for spatial audio. In: *4th international Pure Data Convention 2011, Weimar e Berlin. Proceedings of the 4th international Pure Data Convention 2011*, 2011.

FARIA, R. R. A. AUDIENCE para Pd: uma biblioteca para áudio espacial orientada a cena sonora. In: 10o Congresso de Engenharia de Áudio da AES Brasil, 2012, São Paulo. **Anais do 10o Congresso de Engenharia de Áudio da AES Brasil**. Rio de Janeiro: AES Brasil - Sociedade de Engenharia de Áudio, 2012. p. 98-105.

SILVA, M. J.; FARIA, R. R. A. Aplicação de Wave Field Synthesis para Auralização na Música. In: 11o Congresso de Engenharia de Áudio e 17a Convenção AES Brasil EXPO 2013, 2013, São Paulo. **Anais do 11o Congresso de Engenharia de Áudio da AES Brasil**. Rio de Janeiro: Sociedade de Engenharia de Áudio, 2013. p. 24-30.

SILVA, Márcio José da. Modelagem de um sistema para auralização musical utilizando Wave Field Synthesis. 2014. Dissertação (Mestrado em Processos de Criação Musical) - Escola de Comunicações e Artes, University of São Paulo, São Paulo, 2014. doi:10.11606/D.27.2014.tde-18052015-163521. Acesso em: 07-04-2022.

FARIA, R. R. A.; THOMAZ, L. F.; SOARES, L.; SANTOS, B. T.; ZUFFO, M. K.; ZUFFO, J. A. AUDIENCE - Audio Immersion Experiences in the CAVERNA Digital. In: *The 10th Brazilian Symposium on Computer Music, 2005, Belo Horizonte. Proceedings of the 10th Brazilian Symposium on Computer Music - Current Frameworks for Music Information Representation, 2005.* p. 106-117.

THOMAZ, L. F. FARIA, R. R. A.; ZUFFO, M. K.; ZUFFO, J. A. . Experimentações de espacialização orquestral sobre a arquitetura AUDIENCE. In: 4o. Congresso de Engenharia de Áudio / 10a. Convenção Nacional da AES Brasil, 2006, São Paulo. **Anais do 4o. Congresso de Engenharia de Áudio**, 2006. p. 72-77.

THOMAZ, L. F.; FARIA, R. R. A.; ZUFFO, M. K.; ZUFFO, J. A. Orchestra Spatialization using the AUDIENCE engine. In: **ICMC 2006 - The International Computer Music Conference, 2006, New Orleans. Proceedings, 2006.**

SILVA, M. J.; SCHIAVONI, F. L.; FARIA, R. R. A. Programação dinâmica em Pure Data aplicada a Wave Field Synthesis. In: *AUDIO ENGINEERING SOCIETY. Anais do 12o Congresso de Engenharia de Áudio*. São Paulo, Brasil, 2014.

ZMOLNIG, J. M. *How to write an External for puredata*. Disponível em: <http://mlab.taik.fi/~korayt/kuva/Pd-0.38.4-extended-RC7.app/Contents/Resources/doc/manuals/Externals-HOWTO/pd-externals-HOWTO.pdf>. Acesso em: 26 abr. 2021.