

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

ESTÊVÃO BISSOLI SALEME

**PLAYSEM: UMA PLATAFORMA PARA
RENDERIZAÇÃO DE EFEITOS SENSORIAIS
COMPATÍVEL COM MPEG-V**

**VITÓRIA
2015**

ESTÊVÃO BISSOLI SALEME

**PLAYSEM: UMA PLATAFORMA PARA
RENDERIZAÇÃO DE EFEITOS SENSORIAIS
COMPATÍVEL COM MPEG-V**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Orientador: Prof. Dr. Celso Alberto Saibel Santos.

VITÓRIA
2015

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

S163p Saleme, Estêvão Bissoli, 1983-
PlaySEM : uma plataforma para renderização de efeitos
sensoriais compatível com MPEG-V / Estêvão Bissoli Saleme. –
2015.
108 f. : il.

Orientador: Celso Alberto Saibel Santos.
Dissertação (Mestrado em Informática) – Universidade
Federal do Espírito Santo, Centro Tecnológico.

1. Sistemas multimídia. 2. Estímulos sensoriais. 3. Realidade
virtual. 4. Ferramentas de renderização. 5. Media Player
(Programa de computador). I. Santos, Celso Alberto Saibel. II.
Universidade Federal do Espírito Santo. Centro Tecnológico. III.
Título.

CDU: 004

ESTÊVÃO BISSOLI SALEME

**PLAYSEM: UMA PLATAFORMA PARA
RENDERIZAÇÃO DE EFEITOS SENSORIAIS
COMPATÍVEL COM MPEG-V**

Dissertação submetida ao programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Informática.

Aprovada em _____ de _____ de 2015.

COMISSÃO EXAMINADORA

Prof. Dr. Celso Alberto Saibel Santos
Universidade Federal do Espírito Santo
Orientador

Prof^a. Dra. Roberta Lima Gomes
Universidade Federal do Espírito Santo

Prof. Dr. César Augusto Camillo Teixeira
Universidade Federal de São Carlos

Este trabalho é dedicado a minha família.

Agradecimentos

A Deus, o meu guia e protetor.

À minha esposa Talita, sempre disposta a me ajudar nesta jornada, pelo amor, carinho, companheirismo e compreensão.

Aos meus pais, Roberto e Angela, pelo esforço em me educar e incentivo nos momentos cruciais da minha vida.

Ao meu orientador, Celso, por toda sua colaboração com este trabalho, pela dedicação, confiança e motivação em discutir ideias e materializá-las. Sempre muito ocupado, mas muito solícito e atencioso.

Aos colegas do Laboratório de Pesquisas em Redes e Multimídia, Wancharle, Juliana, Juan, Ricardo, Rafael, dentre outros, pelas contribuições, pelo convívio e aprendizado. Ao professor Almerindo do Instituto Federal de Sergipe pelas discussões acerca de aplicações interativas.

Aos professores do Mestrado em Informática do PPGI/UFES, por compartilhar suas experiências e contribuírem imensamente com meu aprendizado.

Ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, por toda infraestrutura disponibilizada durante o curso para o desenvolvimento deste trabalho.

Ao Instituto Federal do Espírito Santo, pelo incentivo na minha qualificação.

"A mente que se abre a uma nova ideia jamais volta ao seu tamanho original."

(Albert Einstein)

Resumo

No modo tradicional da multimídia somos estimulados basicamente por visão e audição, seja lendo um livro, escutando música, vendo um filme ou visualizando imagens. Um novo horizonte de possibilidades para incrementar a Qualidade de Experiência (QoE) do usuário vem emergindo: as mídias com múltiplos efeitos sensoriais (MulSeMedia - *Multiple Sensorial Media*), que objetivam estimular outros sentidos humanos, como os mecanorreceptores, quimiorreceptores e termorreceptores. Trabalhos recentes têm apontado ganhos significativos de satisfação do usuário nessa direção, que está relacionada ao prazer do usuário a partir de influências da memória na formação da experiência. As soluções existentes para reprodução de efeitos sensoriais embutem o controle dos atuadores em aplicações baseadas em linha de tempo e limitam sua reutilização em outros tipos de aplicação ou diferentes reprodutores de mídia. Esse trabalho apresenta a PlaySEM, uma plataforma que traz uma nova abordagem para a renderização de efeitos sensoriais em ambientes domésticos de modo desacoplado do *Media Player* e compatível com o padrão MPEG-V, considerando requisitos de reutilização e mobilidade. Sobre sua arquitetura são testadas conjecturas, com foco principalmente na questão da abordagem desacoplada do renderizador de efeitos sensoriais.

Palavras-chaves: Múltiplos efeitos sensoriais. MulSeMedia. MPEG-V. Media Player.

Abstract

In the traditional mode of multimedia we're basically stimulated by vision and hearing, be reading a book, listening a music, watching a movie or viewing pictures. A new horizon of possibilities to increase Quality of Experience (QoE) is emerging: the media with Multiple Sensory Effects (MulSeMedia), which aims stimulating other human senses such as mechanoreceptors, chemoreceptors and thermoceptors. Recent works has indicated significant gains in user satisfaction in this order, that is related to the user pleasure from memory influences of the formation of experience. Existing solutions for reproducing sensory effects have embedded control of actuators in timeline-based applications and not support re-use in other application types or different media players. This work presents the PlaySEM, a platform that brings a new approach for rendering sensory effects in home environment of unbound mode of the Media Player and compatible with the MPEG-V standard, considering requirements reuse and mobility. Conjectures are tested about your architecture, focusing mainly on the decoupled approach to render sensory effects.

Key-words: Multiple sensory effects. MulSeMedia. MPEG-V. Media Player.

Lista de ilustrações

Figura 1 – Conceito de efeitos sensoriais.	20
Figura 2 – Esboço do Sensorama à esquerda e o protótipo à direita.	21
Figura 3 – Arquitetura do MPEG-V, V2R (<i>Virtual to Real World</i>) e R2V (<i>Real to Virtual World</i>).	28
Figura 4 – Escopo da parte 2 do MPEG-V destacada na cor esverdeada.	29
Figura 5 – Escopo da parte 3 do MPEG-V destacada na caixa amarela e sua arquitetura.	32
Figura 6 – Modelo de localização para metadados de efeitos sensoriais e sistema de coordenadas de referência.	33
Figura 7 – Modelo de tempo para metadados de efeitos sensoriais.	34
Figura 8 – Mapeamento das intenções do autor dos metadados de efeitos sensoriais com a capacidade dos dispositivos.	36
Figura 9 – Escopo da parte 4 do MPEG-V destacada na caixa amarela.	37
Figura 10 – Escopo da parte 5 do MPEG-V destacada em caixas amarelas.	41
Figura 11 – Interface do SEVino	46
Figura 12 – Interface do SMURF	47
Figura 13 – Interface do RoSE Studio e seus componentes	48
Figura 14 – SEMP e o sistema amBX	49
Figura 15 – Interface do SESim e sua janela com informações dos dispositivos	50
Figura 16 – Interface do Sensorama	51
Figura 17 – Interface do Sensible Media Simulator	52
Figura 18 – Philips amBX Gaming PC peripherals	54
Figura 19 – Mad Catz Cyborg Gaming Lights	55
Figura 20 – Dale Air Vortex Activ	55
Figura 21 – CAVE System	56
Figura 22 – Arquitetura conceitual da plataforma PlaySEM.	63
Figura 23 – Processo de ligação JAXB.	66

Figura 24 – Processo de criação da biblioteca MPEG <i>Metadata</i>	67
Figura 25 – Estrutura e relacionamentos dos componentes da plataforma PlaySEM.	68
Figura 26 – Camadas da plataforma PlaySEM adaptadas na arquitetura da parte 3 do MPEG-V proposta por Choi e Kim.	70
Figura 27 – <i>Home Server</i> à esquerda e <i>SMMD Media Controller</i> à direita	71
Figura 28 – Diagrama de casos de uso do software SE <i>Video Player</i>	72
Figura 29 – Interface do software SE <i>Video Player</i>	74
Figura 30 – Procura por dispositivo SE <i>Renderer</i> a partir do SE <i>Video Player</i> por meio de interface UPnP.	74
Figura 31 – Descrição das capacidades dos dispositivos.	75
Figura 32 – Diagrama de atividades – Execução de Vídeo.	76
Figura 33 – Transformação de anotação de efeito de luz para comandos com “fade in” de 1 segundo.	77
Figura 34 – Diagrama de casos de uso do software SE <i>Renderer</i>	78
Figura 35 – Cling Workbench - visualização dos serviços e variáveis do SE <i>Renderer</i>	79
Figura 36 – Diagrama de atividades – Preparação de efeitos sensoriais.	81
Figura 37 – Diagrama de atividades – Execução de ações.	82
Figura 38 – Hardwares no ambiente real do usuário.	84
Figura 39 – Diagrama de sequência – Selecionar dispositivo.	86
Figura 40 – Diagrama de sequência – Abrir vídeo.	87
Figura 41 – Diagrama de sequência – Auto extração de luz.	88
Figura 42 – Diagrama de sequência – Play.	89
Figura 43 – Etapas do método hipotético-dedutivo.	91
Figura 44 – Tempo de execução dos comandos x Variação da latência.	96
Figura 45 – Latência em relação ao tempo total de processamento.	96
Figura 46 – Diferença de tempo entre o relógio do <i>Player</i> e do SE <i>Renderer</i> em milissegundos.	97
Figura 47 – Conjunto de interações e ações especificadas em cada contexto.	99
Figura 48 – À esquerda um esboço do ambiente e seus componentes. À direita é mostrado a sequência de efeitos no ambiente real.	99

Lista de quadros

Quadro 1 – Comparativo dos trabalhos relacionados compostos por ferramentas de autoria, reprodução e simulação de efeitos sensoriais compatíveis com MPEG-V.	58
Quadro 2 – Ferramentas utilizadas no processo de desenvolvimento da plataforma PlaySEM.	64
Quadro 3 – Função dos componentes da arquitetura da plataforma PlaySEM. . . .	69
Quadro 4 – Descrição dos serviços expostos pelo SE <i>Renderer</i>	80
Quadro 5 – Descrição das variáveis expostas pelo SE <i>Renderer</i>	80
Quadro 6 – Hardwares acoplados ao SE <i>Renderer</i> para renderização de efeitos de luz, vento e vibração.	83

Lista de tabelas

Tabela 1 – Tempo médio de latência na execução dos serviços do SE <i>Renderer</i> . . .	95
Tabela 2 – Diferença de tempo entre o relógio do <i>Player</i> e do SE <i>Renderer</i> em milissegundos.	97
Tabela 3 – Tempo de pré-processamento dos <i>scripts</i> MPEG-V usados no estudo de caso.	100
Tabela 4 – Tempo de pré-processamento de <i>script</i> MPEG-V ampliado com 90 efeitos sensoriais para aplicação orientada a evento.	100

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i> (Interface de Programação de Aplicações)
CAVE	<i>Cave Automatic Virtual Environment</i> (Caverna Digital)
CIDL	<i>Control Information Description Language</i> (Linguagem de Descrição de Controle da Informação)
DCDV	<i>Device Capability Description Vocabulary</i> (Vocabulário de Descrição de Capacidade do Dispositivo)
DCV	<i>Device Command Vocabulary</i> (Vocabulário de Comando de Dispositivo)
EBNF	<i>Extended Backus–Naur Form</i> (Formalismo de Backus-Naur Estendido)
ECB	<i>Entity, Controller, Boundary</i> (Entidade, Controlador, Fronteira)
ECU	<i>Electronic Control Unit</i> (Unidade de Controle Eletrônica)
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
IIDL	<i>Interaction Interface Description Language</i> (Linguagem de Descrição de Interface da Interação)
JAXB	<i>Java Architecture for XML Binding</i> (Arquitetura Java para Ligação XML)
JRE	<i>Java Runtime Environment</i> (Ambiente de Execução Java)
LGPL	<i>GNU Lesser General Public License</i>
MP	<i>Media Player</i>
MPEG-2-TS	<i>Transport Stream</i> (Fluxo de Transporte)
OO	<i>Object Oriented</i> (Orientado a Objeto)
QoE	<i>Quality of Experience</i> (Qualidade de Experiência)
RoSE	<i>Representation of Sensory Effects</i> (Representação dos Efeitos Sensoriais)
RW	<i>Real World</i> (Mundo Real)
SAPV	<i>Sensor Adaptation Preference Vocabulary</i> (Vocabulário de Preferência de Adaptação do Sensor)

SCDV	<i>Sensor Capability Description Vocabulary</i> (Vocabulário de Descrição de Capacidade do Sensor)
SDK	<i>Software Development Kit</i> (Kit de Desenvolvimento de Software)
SE	<i>Sensory Effect</i> (Efeito Sensorial)
SEDL	<i>Sensory Effect Description Language</i> (Linguagem de Descrição de Efeito Sensorial)
SEM	<i>Sensory Effects Metadata</i> (Metadados de Efeitos Sensoriais)
SEPV	<i>Sensory Effect Preference Vocabulary</i> (Vocabulário de Preferência de Efeito Sensorial)
SER	<i>Sensory Effects Renderer</i> (Renderizador de Efeitos Sensoriais)
SEV	<i>Sensory Effect Vocabulary</i> (Vocabulário de Efeito Sensorial)
SEVino	<i>Sensory Effect Video Annotation</i> (Anotação de Vídeo com Efeito Sensorial)
SIV	<i>Sensed Information Vocabulary</i> (Vocabulário de Informação Percebida)
SMCS	<i>Sensible Media Control System</i> (Sistema de Controle de Mídia Sensível)
SMURF	<i>Sensible Media aUthoRing Factory</i> (Fábrica de Autoria de Mídia Sensível)
UDA	<i>UPnP Device Architecture</i> (Arquitetura de Dispositivo UPnP)
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)
UPnP	<i>Universal Plug and Play</i>
V2V	<i>Virtual to Virtual</i> (Virtual para Virtual)
VW	<i>Virtual World</i> (Mundo Virtual)
XML	<i>eXtensible Markup Language</i> (Linguagem de Marcação Extensível)
XMM	<i>XML Metadata Management</i> (Gestão de Metadados XML)
XSD	<i>XML Schema Definition</i> (Esquema de Definição XML)

Sumário

1	INTRODUÇÃO	19
1.1	Contexto e motivação	19
1.2	Problemas relacionados	22
1.3	Objetivo e contribuições	23
1.4	Organização do trabalho	25
2	O PADRÃO ISO/IEC 23005 - MPEG-V	27
2.1	MPEG-V Parte 1 - Arquitetura	27
2.2	MPEG-V Parte 2 - Informações de Controle	29
2.3	MPEG-V Parte 3 – Efeitos Sensoriais	31
2.4	MPEG-V Parte 4 – Características de Objetos do Mundo Virtual	36
2.5	MPEG-V Parte 5 – Formato de Dados para Dispositivos de Interação	40
2.6	MPEG-V Parte 6 – Ferramentas e Tipos Comuns	42
2.7	MPEG-V Parte 7 – Conformidade e Software de Referência	43
2.8	Conclusão	43
3	TRABALHOS RELACIONADOS	45
3.1	Ferramentas de autoria de efeitos sensoriais	45
3.1.1	SEVino	45
3.1.2	SMURF	46
3.1.3	RoSE Studio	47
3.2	<i>Media players</i> de efeitos sensoriais	48
3.2.1	SEMP	49
3.2.2	SESim	50
3.2.3	Sensorama	51
3.2.4	Sensible Media Simulator	52
3.3	Atuadores para renderizadores de efeitos sensoriais	53

3.3.1	amBX Gaming PC peripherals	53
3.3.2	Cyborg Gaming Lights for PC	54
3.3.3	Vortex Activ	55
3.3.4	CAVE System	56
3.4	Conclusão	57
4	PROJETO E IMPLEMENTAÇÃO DA PLATAFORMA PLAYSEM	62
4.1	Metodologia, ferramentas e tecnologias	64
4.1.1	Cling UPnP Core	64
4.1.2	VLCJ	65
4.1.3	MPEG Metadata	65
4.2	A Plataforma PlaySEM compatível com MPEG-V	67
4.2.1	Arquitetura do sistema	67
4.2.2	PlaySEM SE <i>Video Player</i>	71
4.2.3	PlaySEM SE <i>Renderer</i>	77
4.2.3.1	Software	77
4.2.3.2	Hardware	82
4.3	Adaptação de <i>Media Players</i>	84
4.3.1	Recomendações gerais	84
4.3.2	Diagramas para troca de mensagens	85
4.4	Conclusão	89
5	EXPERIMENTOS E RESULTADOS	91
5.1	Metodologia	91
5.2	Resultados da avaliação	94
5.2.1	Uso em rede	94
5.2.2	Impacto na perda de sincronização	96
5.2.3	Estudo de caso: uso em aplicação interativa baseada em evento	97
6	CONCLUSÃO E TRABALHOS FUTUROS	101
6.1	Trabalhos futuros	103

Referências	105
-----------------------	-----

1 Introdução

1.1 Contexto e motivação

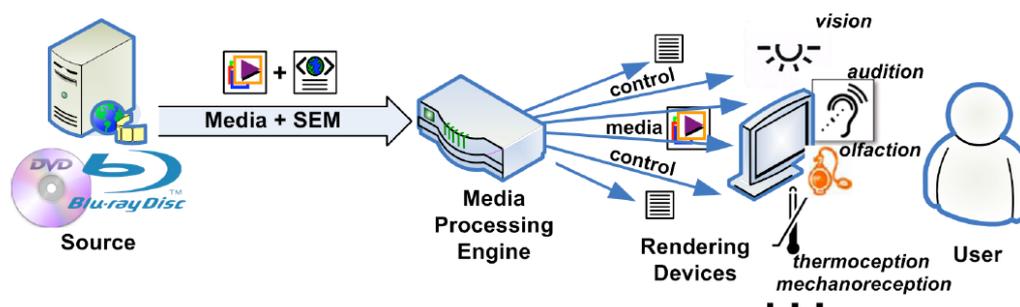
No modo tradicional da multimídia somos estimulados basicamente por visão e audição, seja lendo um livro, escutando música, vendo um filme ou visualizando imagens (GHINEA et al., 2014; PREDA, 2011). A popularização de mídias e dispositivos capazes de proporcionar conteúdo audiovisual em formato 3D na década passada, tanto no cinema quanto no ambiente doméstico, foi importante para elevar os níveis de imersão do usuário no ambiente virtual (MÖLLER; RAAKE, 2014). Adicionalmente, um novo horizonte de possibilidades para incrementar a Qualidade de Experiência (*Quality of Experience* - QoE) do usuário vem emergindo: as mídias com múltiplos efeitos sensoriais (MulSeMedia - *Multiple Sensorial Media*) (GHINEA et al., 2014). Essas apresentações objetivam estimular outros sentidos humanos, como os mecanorreceptores, sensíveis às variações de movimento e força, quimiorreceptores, sensíveis à presença substâncias químicas, e termorreceptores, sensíveis às mudanças de temperatura (KIM; HAN, 2014; WALTL; TIMMERER; HELLWAGNER, 2009). Esses sensores podem ser ativados por meio de efeitos de luz, vibração, vento, aroma, etc.

Pesquisas recentes na área de multimídia concentraram esforços na direção da melhoria da QoE (EBRAHIMI, 2009; MÖLLER; RAAKE, 2014; RAINER et al., 2012; WALTL; TIMMERER; HELLWAGNER, 2010). Surgiu uma nova perspectiva de qualidade centrada no usuário, envolvendo não somente a percepção subjetiva, mas também o comportamento e necessidades do usuário paralelo à Qualidade de Serviço (*Quality of Service* - QoS), que objetiva capturar características relacionadas à performance de sistemas de telecomunicações (EBRAHIMI, 2009; MÖLLER; RAAKE, 2014). Experimentos recentes de QoE no campo de efeitos sensoriais (RAINER et al., 2012; WALTL; TIMMERER; HELLWAGNER, 2010; YUAN; GHINEA; MUNTEAN, 2015) tem apontado ganhos significativos de satisfação do usuário. Watl, Timmerer e Hellwagner (2010) perceberam que a mídia acompanhada de efeitos sensoriais melhora a opinião dos usuários mesmo com a superioridade na qualidade visual do vídeo da mídia sem efeitos sensoriais.

Para entender como de fato os múltiplos efeitos sensoriais proporcionam a imersão do usuário no ambiente virtual, a Figura 1, reproduzida a partir do artigo de Watl, Timmerer e Hellwagner (2009), ilustra os conceitos desse cenário. A mídia e o arquivo correspondente de metadados de efeitos sensoriais (SEM – *Sensory Effect Metadata*) podem ser obtidos a partir de mídias de armazenamento, como por exemplo DVD (*Digital Versatile Disc*), *Blu-Ray* ou algum tipo de serviço online. O motor de processamento da mídia (*Media Processing Engine*) atua como um mediador, sendo responsável por interpretar os

recursos da mídia, adaptando tanto a mídia quanto o SEM às capacidades dos dispositivos que renderizarão os efeitos sensoriais e às preferências dos usuários. O ambiente do usuário é estendido com dispositivos capazes de estimular outros sentidos humanos além da visão e audição, como por exemplo, cadeiras de vibração, ventiladores, aromatizadores, etc (MÖLLER; RAAKE, 2014; WALTL; TIMMERER; HELLWAGNER, 2009).

Figura 1 – Conceito de efeitos sensoriais.



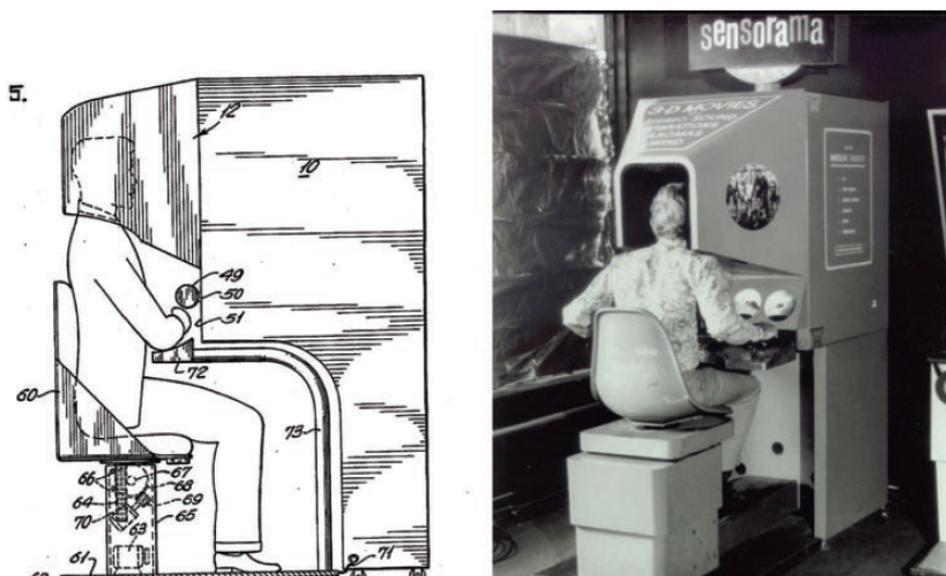
Fonte: (WALTL; TIMMERER; HELLWAGNER, 2009)

Ghinea, Andres e Gulliver (2011) definem imersão como uma experiência realista, vista como condição necessária para criação de “presença”, que está relacionada ao prazer do usuário a partir de influências da memória na formação da experiência. No contexto de ambientes virtuais, a ideia de estar presente refere-se a experimentar o ambiente gerado por computador, em vez da localidade física real (WITMER; SINGER, 1998). Witmer e Singer (1998) conceituam imersão como um estado psicológico caracterizado por perceber-se envolvido, incluído e interagindo com um ambiente que fornece um fluxo contínuo de estímulos e experiências. Quanto maior o nível de imersão, maior o nível de presença. De modo mais conciso Preda (2011) apresenta imersão como uma abstração do mundo físico por meio do “mundo da informação”, que podem ser filmes, livros, novelas, histórias, jogos, etc.

A origem de simuladores para criar imersão por meio de múltiplos efeitos sensoriais em ambientes virtuais é da década de 60 (GHINEA et al., 2014), com o desenvolvimento do simulador Sensorama, patenteado por Morton Heilig como “um aparato para estimular os sentidos de um indivíduo para simular uma experiência realística” (GHINEA; ANDRES; GULLIVER, 2011). A Figura 2 mostra o simulador, que consiste em um protótipo máquina em que o usuário está imerso em imagens 3D, odores, sons estéreo, ventos e vibrações.

A inclusão de múltiplos efeitos sensoriais tem sido essencial para aumentar a imersão e a presença no ambiente do usuário (GHINEA; ANDRES; GULLIVER, 2011). É por meio da percepção das sensações do usuário que isso ocorre. Para Ghinea et al. (2014) a percepção é o resultado de um conjunto complexo de processos, em que os sensores biológicos encaminham sinais elétricos estruturados para o cérebro (exceto

Figura 2 – Esboço do Sensorama à esquerda e o protótipo à direita.



Fonte: ([GHINEA; ANDRES; GULLIVER, 2011](#))

quimiorreceptores específicos), que por sua vez forma inconscientemente os padrões de sensações. Essa estruturação ajuda a determinar se uma entrada sensorial é realística ou não. [Möller e Raake \(2014\)](#) consideram que a percepção passa por 2 estágios antes de ser finalmente concluída, o primeiro é a conversão do estímulo por meio do respectivo órgão sensorial em sinais neurais e a segunda é o processamento e transmissão desses sinais neurais no sistema nervoso central para o córtex, resultando no aparecimento de percepções específicas no mundo perceptual da pessoa. Em outro trabalho, [Ghinea, Andres e Gulliver \(2011\)](#) são mais pragmáticos ao afirmarem que a percepção humana é inerentemente multissensorial envolvendo aspectos visuais, auditivos, táteis, olfativas, gustativos, etc.

A imersão por múltiplos efeitos sensoriais não está restrita à indústria do entretenimento. [Ghinea et al. \(2014\)](#) em trabalho recente sobre o estado da arte de MulSeMedia fazem menção aos relatos sobre o potencial uso terapêutico para pessoas com necessidades especiais, que vão desde a dificuldade de aprendizagem até o autismo, mal de Alzheimer e a demência. Outra área destacada no trabalho é a indústria de jogos que pode se beneficiar dos múltiplos efeitos sensoriais para aumentar a sensação de presença tornando os jogos mais envolventes para os usuários. Baseado na ideia de aproveitar a infraestrutura de um carro moderno, composto por dispositivos capazes de renderizarem efeitos sensoriais e diversos sensores, [Kim e Joo \(2014\)](#) propuseram um sistema para levar entretenimento para automóveis. Tais iniciativas revelam que a MulSeMedia pode ser explorada em diversas áreas.

De acordo com os dicionários [Cambridge \(2015\)](#) e [Oxford \(2015\)](#) renderizar significa entregar alguma coisa a alguém, como por exemplo, um serviço. Desse modo, no contexto

de efeitos sensoriais, renderizar quer dizer entregar os efeitos sensoriais como luz, vento, vibração, etc, no ambiente do usuário. O padrão MPEG-V (CHOI; KIM, 2012; HAN; YOON, 2012; HAN; PREDA, 2012; KIM; HAN; YOON, 2012; KIM; JOO; CHOI, 2013; YOON; HAN, 2012), também conhecido como ISO/IEC 23005, provê uma arquitetura e especifica representações de informação para conectar o mundo virtual e mundo real, incluindo metadados de efeitos sensoriais.

1.2 Problemas relacionados

Soluções heterogêneas para resolver o mesmo problema são desenvolvidas a todo instante tanto pela indústria quanto pela academia. Um exemplo é relatado no trabalho de (CHO, 2010), em que parques temáticos envolvendo alta tecnologia, como Disney Land e Universal Studios, produzem conteúdo com múltiplos efeitos sensoriais de modo distinto, inviabilizando a reutilização de produções similares em ambientes diferentes. Esse é um problema típico de integração de aplicações em ambientes de interação avançada, em que tanto a entrada como saída de dados não utilizam um formato comum para interoperabilidade (SANTOS; SALEME; ANDRADE, 2015). O ECIS (*European Committee for Interoperable Systems*) diz que a interoperabilidade é um dos pilares da indústria de TIC e que um dispositivo que não pode interoperar com outros produtos que o cliente espera interoperar não tem valor (ECIS, 2015). No cenário de múltiplos efeitos sensoriais, é importante que sejam desenvolvidas ferramentas capazes de prover um acesso universal à mídia, permitindo a autoria dos efeitos sensoriais pelo próprio produtor da mídia. Para isso, a utilização de um padrão para integrar o mundo real ao mundo virtual e vice-versa se faz necessária.

Além do problema da interoperabilidade, o acoplamento de renderizadores de efeitos sensoriais ao *Media Player* não permite a reutilização e mobilidade dos dispositivos físicos sobre camadas de apresentação distintas. Por exemplo, se o usuário deseja assistir um filme com múltiplos efeitos sensoriais a partir de uma *Smart TV* ou a partir de um *smartphone* com saída para monitores, a solução dos *Media Players* de efeitos sensoriais atuais é ineficiente, pois concentra tanto a reprodução do vídeo quanto o processamento dos metadados de efeitos sensoriais em um único componente. Para hardwares com recursos limitados a complexidade de processar vídeos, metadados de efeitos sensoriais e controle dos atuadores em tempo real, configura-se um fator restritivo para utilização de mídias com múltiplos efeitos sensoriais. Soluções que desacoplem a camada de apresentação da camada de renderização de efeitos sensoriais permitem a reutilização da última em *Media Players* distintos, desde que ofereçam uma interface para comunicação. No entanto, há de se avaliar se essa abordagem não traz atrasos relevantes em tempo de resposta e se essa arquitetura tem impacto com relação à dessincronização dos efeitos sensoriais x *media players*. Parâmetros de medição precisam ser discutidos para avaliar a eficiência da

plataforma.

O acoplamento de renderizadores de efeitos sensoriais aos *Media Players*, que reproduzem conteúdo audiovisual associado às descrições de efeitos sensoriais, também dificulta a adoção de múltiplos efeitos em outras áreas como medicina, jogos e educação. Isso porque o modelo atual induz a entrada dos metadados de efeitos sensoriais a partir de *Media Players* com uma única linha de tempo, diferente dos modelos geralmente desenvolvidos para aplicações nas áreas relatadas em que a mídia principal é baseada em eventos. Por exemplo, quando ocorre uma explosão em um jogo, espera-se ações de múltiplos efeitos sensoriais de luz e vibração no ambiente do usuário para simulá-la. Esse tipo de evento pode ocorrer constantemente em uma aplicação mas precisam ser programados manualmente. Além disso, são necessárias adaptações para atuadores de diferentes fabricantes. Fica evidente mais uma vez a necessidade de renderizadores de efeitos sensoriais que possam ser reaproveitados e compatíveis com padrões para descrição dos metadados de efeitos sensoriais nas aplicações.

Por último, a ausência de hardwares abertos e de baixo custo capazes de renderizar efeitos sensoriais em níveis variáveis de intensidades no ambiente doméstico também é uma realidade. Os atuadores existentes são equipamentos comerciais para uso doméstico e em parques temáticos de alta tecnologia, no entanto, ainda possuem um alto custo. Em relação aos *Media Players* encontrados na literatura, somente um é capaz de renderizar efeitos sensoriais efetivamente no ambiente do usuário, porém é suportado apenas no ambiente operacional Windows, restringindo o acesso universal. Novos reproduzidores de mídias com suporte a múltiplos efeitos sensoriais capazes de serem executados em ambientes operacionais distintos (Linux, Mac OSX, etc) são necessários.

1.3 Objetivo e contribuições

Em linhas gerais, esse trabalho propõe uma nova abordagem para a renderização de efeitos sensoriais de modo desacoplado do *Media Player* usando a comunicação UPnP e compatível com o padrão MPEG-V. A solução permite que o renderizador de efeitos sensoriais, bem como os atuadores, possa ser reaproveitado em outras interfaces de apresentação, tais como jogos, aplicações interativas e diferentes *Media Players*. A partir dessa abordagem a plataforma denominada PlaySEM, um acrônimo de Reprodução (Play) de Metadados de Efeitos Sensoriais (SEM – Sensory Effect Metadata), foi desenvolvida. Ela fornece um *Video Player* capaz de ler metadados de efeitos sensoriais no formato MPEG-V e se comunicar com o renderizador de efeitos sensoriais, que por sua vez transforma os metadados em comandos para os atuadores de luz, vibração e vento.

Todos os softwares e hardwares do PlaySEM estão disponibilizados no formato aberto, propiciando a oportunidade de reprodução de mídias anotadas com efeitos sensoriais

a um baixo custo associado. As principais contribuições desse trabalho estão resumidas nos seguintes pontos:

- Renderizador de efeitos sensoriais reutilizável. O módulo renderizador de efeitos sensoriais é desacoplado no *Video Player*, facultando o uso de outros *Media Players* ou aplicações multimídia, como por exemplo jogos. O protocolo UPnP usado em seu núcleo permite a descoberta dinâmica, descrição, controle, apresentação e envio de eventos a partir de dispositivos remotos de modo padrão, assegurando que dispositivos de diferentes fornecedores se comuniquem com o renderizador. Essa característica do SE *Renderer* concede ao usuário a possibilidade de deslocar os dispositivos para reutilização em outros ambientes.
- SE *Video Player* compatível com o renderizador de efeitos sensoriais. O software é capaz de ler arquivos de metadados de efeitos sensoriais descritos no padrão MPEG-V com extensão XML ou SEM e funciona como um *Media Player* tradicional na ausência do renderizador de efeitos sensoriais. Controles de estado da mídia, como *play*, *pause*, *stop* ou avançar e retroceder, são sincronizados constantemente com o renderizador fazendo com que os efeitos atuem no ambiente do usuário consoante a linha de tempo. Há também utilidade para o entendimento de como realizar adaptações de *Media Players* ou aplicações ao renderizador de efeitos sensoriais e reaproveitá-lo.
- Biblioteca Java de formato aberto MPEG *Metadata* (Metadados MPEG) para transformação de XML em objeto Java. Arquivos de metadados de efeitos sensoriais ou a descrição das capacidades do dispositivo são descritos em XML seguindo o padrão MPEG-V e precisam ser lidos pela aplicação. Para facilitar a ligação entre o XML e objetos de uma aplicação Java, foi desenvolvida a biblioteca MPEG *Metadata*, que contém todos os objetos possíveis do padrão MPEG-V (parte 1 até 5) e parte dos padrões MPEG-7 e MPEG-21, possibilitando o desempacotamento automático do XML na aplicação por meio da API JAXB - *Java Architecture for XML Binding*. A principal vantagem da utilização da biblioteca em aplicações que fazem uso do MPEG-V é a possibilidade de reusá-la sem que o desenvolvedor se preocupe com o mapeamento XML/Objeto. Ela é utilizada tanto no software SE *Video Player* quanto no SE *Renderer*.
- Plataforma simples e de baixo custo para a simulação de efeitos sensoriais. Para a produção de efeitos de luz no ambiente é utilizada uma fita de LED endereçável, para vento são dois ventiladores PC *Fan*, e para vibração dois motores de vibração de celulares. O aparato tecnológico para atuação dos efeitos sensoriais no ambiente do usuário, excetuando a fita de led endereçável, são peças descartadas de outros

dispositivos eletrônicos, com custo acessível. Um microcontrolador Arduino, componente do renderizador de efeitos sensoriais, é responsável por receber comandos e acionar os respectivos atuadores para efeitos de luz, vento e vibração. O baixo custo associado proporciona um aumento potencial de acesso de qualquer usuário à plataforma.

- Plataforma de software livre que permite reproduzir, simular e renderizar efeitos sensoriais de luz, vento e vibração no ambiente real por meio de um padrão, indicando atributos como intensidade, posição, duração, etc. Em conformidade com o padrão MPEG-V, que possibilita a descrição de efeitos sensoriais de mídias, as ferramentas desenvolvidas nesse trabalho permitem a reprodução da mídia por meio do software PlaySEM SE *Video Player* e a renderização ou simulação (habilitando o modo de depuração) dos efeitos sensoriais por meio do software (e hardwares) do PlaySEM SE *Renderer* no ambiente real do usuário, proporcionando imersão. A característica de código aberto da plataforma, fornece liberdade de execução, de redistribuição, de estudo ou modificação das ferramentas, proporcionando incentivo ao desenvolvimento tecnológico.

1.4 Organização do trabalho

Essa dissertação está organizada da seguinte forma:

- O capítulo 2 fornece conceitos sobre todas as partes do padrão MPEG-V (ISO 23005), que define representações para conectar o mundo real e virtual. A parte 2 (ISO 23005-2), que inclui a descrição sobre capacidades dos dispositivos e informações de preferência do usuário, e a parte 3 (ISO 23005-3) que contém uma estruturação para descrição de informações sobre efeitos sensoriais, tem a sua arquitetura vastamente explorada permitindo a compreensão sobre como os metadados de efeitos sensoriais são escritos e como devem ser adaptados e interpretados pelos motores de renderização.
- O capítulo 3 apresenta uma revisão da literatura sobre os trabalhos relacionados. São apresentadas ferramentas de autoria de metadados de efeitos sensoriais, *Media Players* com capacidade de reprodução ou simulação de efeitos sensoriais compatíveis com o MPEG-V e os atuadores para renderização de efeitos sensoriais utilizados nos principais *Media Players*. Ao final, os trabalhos são comparados e discutidos considerando os objetivos de cada um, o modo de comunicação e possibilidades de reutilização, incluindo a proposta desse trabalho.
- O capítulo 4 descreve a arquitetura e implementação da plataforma PlaySEM. São apresentadas a metodologia para o desenvolvimento da plataforma bem como as

ferramentas e tecnologias empregadas no processo. A arquitetura é decomposta em componentes menores para entendimento da sua estrutura. As funcionalidades e comportamentos dos sistemas que compõe a plataforma são detalhados. Os hardwares para renderização de efeitos sensoriais são descritos e caracterizados. Ao final são apresentadas recomendações para adaptações de outros *Media Players* ao renderizador de efeitos sensoriais do PlaySEM.

- O capítulo 5 mostra a avaliação da plataforma PlaySEM considerando a abordagem desacoplada. São conduzidos experimentos para verificar se há atrasos relevantes na comunicação e o impacto na dessincronização. Ademais, é apresentado um caso de estudo de uma aplicação interativa baseada em evento com o intuito de avaliar sua aderência ao modelo desacoplado.
- O capítulo 6 resume as contribuições dessa pesquisa apresentando a conclusão. São discutidos também trabalhos futuros que podem adicionalmente contribuir para pesquisas no campo de efeitos sensoriais.

2 O padrão ISO/IEC 23005 - MPEG-V

Sob a perspectiva do usuário, interfaces divertidas, intuitivas, eficientes e eficazes são cruciais para o uso e aceitação maciça dessas interfaces. Nesse sentido, é indispensável o emprego de ferramentas e de metodologias apropriadas (KIM; HAN, 2014). É desejável uma conexão bem definida entre o mundo real e virtual para que os estímulos produzidos por meio da interface e do ambiente resultem numa efetiva melhoria na Qualidade da Experiência do Usuário (QoE) durante sua interação. O padrão ISO/IEC 23005, também conhecido como MPEG-V, surgiu nesse contexto. Ele provê uma arquitetura e especifica representações de informação para conectar o mundo virtual (ex. jogos, simuladores e aplicações) e mundo real (ex. sensores e atuadores) (KIM; HAN, 2014; KIM; HAN; YOON, 2012). Com foco na questão da interoperabilidade, o padrão envolve especificações sobre capacidade de dispositivos, efeitos sensoriais, características de objetos do mundo virtual e formato de dados para interação entre dispositivos.

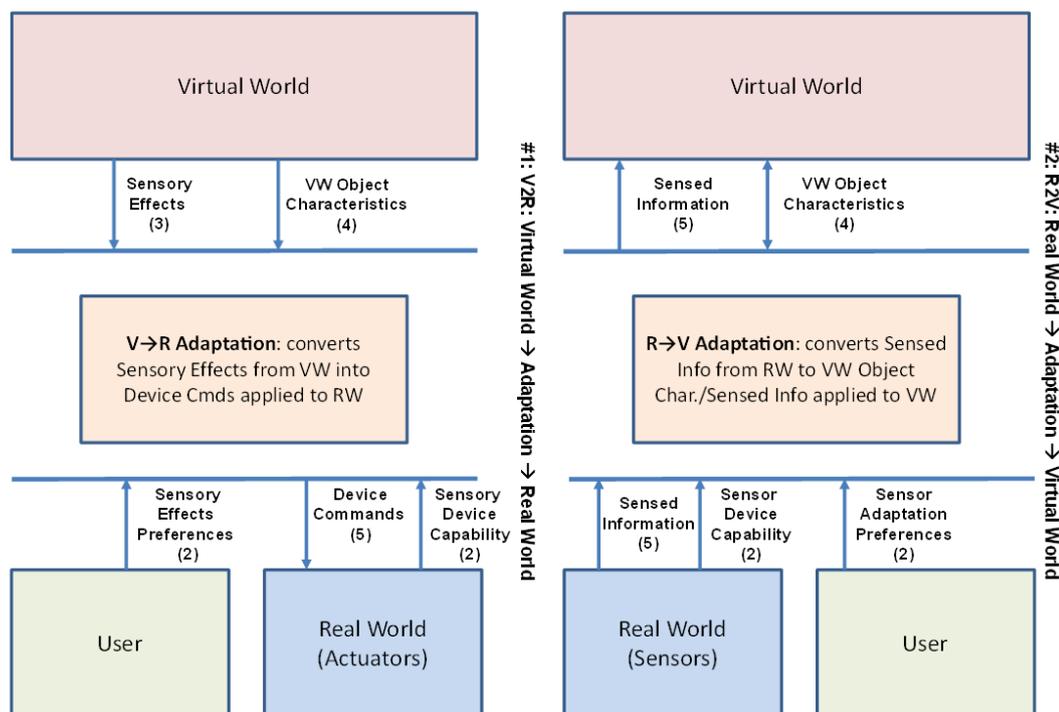
Vários modelos de domínio em que conteúdos audiovisuais podem ser associados com efeitos sensoriais podem ser beneficiados da definição proposta na norma (KIM; HAN, 2014). Como exemplo, pode-se distribuir um filme (conteúdo audiovisual) composto por efeitos sensoriais por meio de *broadcast* em que os receptores podem nem sempre contar com atuadores do mesmo fabricante, mas podem se comunicar com eles por meio de um protocolo comum, o MPEG-V, para acionar os efeitos nesses dispositivos. Outro cenário, não menos oportuno, seria o uso do padrão em jogos, de forma semelhante ao cenário de *broadcast*: um conteúdo (jogo multimodal e multissensorial) sendo distribuído para diversificados atuadores e configurações próprias de cada ambiente disponível para o usuário.

Nesse capítulo é apresentada a arquitetura do MPEG-V e todas as partes que o compõem: parte 1 – arquitetura; parte 2 – informações de controle; parte 3 – efeitos sensoriais; parte 4 – características de objetos do mundo virtual; parte 5 – formato de dados para dispositivos de interação; parte 6 – ferramentas e tipos comuns e; parte 7 – conformidade e software de referência. O texto dá uma ênfase maior na parte 3, a qual está relacionada à maioria dos conceitos abordados nesse trabalho.

2.1 MPEG-V Parte 1 - Arquitetura

A especificação da arquitetura proposta para o MPEG-V é apresentada no documento que descreve o padrão ISO/IEC 23005-1 (KIM; HAN, 2014). A Figura 3 ilustra a representação dessa arquitetura, bem como o relacionamento entres as partes que formam o conjunto de padrões MPEG-V.

Figura 3 – Arquitetura do MPEG-V, V2R (*Virtual to Real World*) e R2V (*Real to Virtual World*).



Fonte: (KIM; HAN; YOON, 2012)

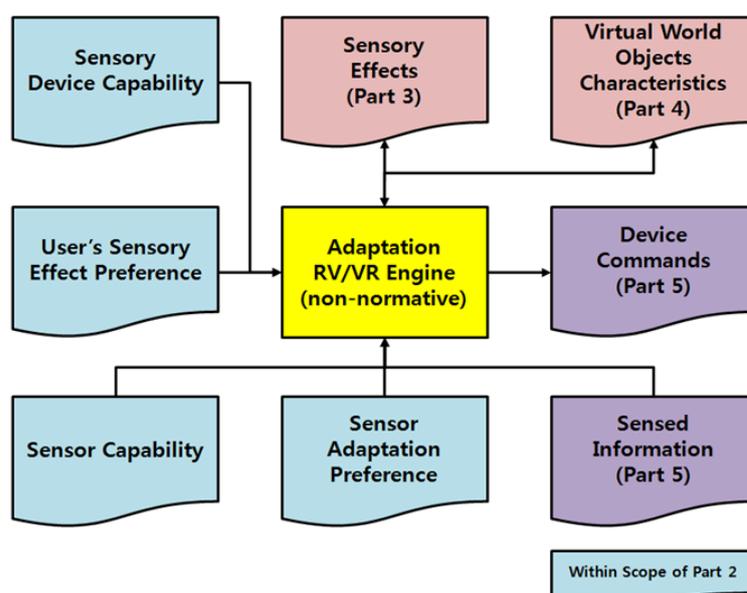
Do Mundo Virtual (*Virtual World* - VW) para o Mundo Real (*Real World* - RW), informações dos efeitos sensoriais (parte 3) e características de objetos virtuais (parte 4) podem ser encaminhadas para o ambiente físico do usuário, para que sejam reproduzidas. O motor de adaptação (opcional), que serve para converter descrições de efeitos sensoriais (parte 3) em comandos para dispositivos (parte 5), configurar preferências de efeitos sensoriais do usuário (parte 2) e descrever a capacidades dos dispositivos (parte 2), não está no âmbito da norma, preocupada somente com a representação das informações. Os atuadores podem ser compatíveis diretamente com a parte 3, não sendo necessária a conversão de um efeito em N comandos.

Do Mundo Real para o Mundo Virtual, informações obtidas do ambiente real do usuário (parte 5) e suas preferências (parte 2) podem ser encaminhadas para o ambiente virtual. Por exemplo, informações de temperatura do ambiente real podem ser reproduzidas no ambiente virtual. Há ainda a possibilidade de interoperar informações entre mundos virtuais distintos (V2V), a partir da parte 4, como por exemplo, tornar compatíveis informações de características visuais de um avatar de um dado jogo com outro avatar em um jogo distinto. O motor de adaptação desse cenário também não é abrangido pela norma.

2.2 MPEG-V Parte 2 - Informações de Controle

A parte 2 do MPEG-V (ISO/IEC 23005-2) aborda aspectos normativos das informações de controle, incluindo descrição capacidade do dispositivo e informações de preferência do usuário (HAN; YOON, 2012). Seu escopo, representado na Figura 4, cobre interfaces entre o motor de adaptação, descrições de capacidade de sensores e atuadores no mundo real e informações de preferências dos usuários sobre esses dispositivos. A sintaxe e a semântica definidas na norma estabelecem o mecanismo para prover a interoperabilidade entre dispositivos e sistemas. Dessa forma, a parte 2 provê meios de fornecer informações extras para o motor de adaptação (que não é normativo) interagir com as demais partes do MPEG-V prevendo a capacidade disponível e as preferências configuradas pelo usuário.

Figura 4 – Escopo da parte 2 do MPEG-V destacada na cor esverdeada.



Fonte: (HAN; YOON, 2012)

Essa parte especifica uma linguagem de descrição de controle da informação (CIDL - *Control Information Description Language*) baseada em XML que permite descrever uma estrutura básica de informações de controle (KIM; HAN, 2014). Como parte da CIDL, quatro vocabulários apoiam a representação das informações:

- Vocabulário para descrição das capacidades dos dispositivos (DCDV - *Device Capability Description Vocabulary*). Exemplo: a velocidade máxima de vento suportada pelo dispositivo e o número de níveis de rotação;
- Vocabulário para descrição das capacidades dos sensores (SCDV - *Sensor Capability Description Vocabulary*). Exemplo: a temperatura mínima e máxima suportada por um sensor de temperatura;

- Vocabulário de preferências de efeitos sensoriais do usuário (SEPV - *User's Sensory Effect Preference Vocabulary*). Exemplo: o usuário quer restringir o efeito sensorial de vibração ao nível médio, pois acredita que o nível máximo pode ser perigoso para uma criança;
- Vocabulário para preferência do usuário para adaptação dos sensores (SAPV - *Sensor Adaptation Preference Vocabulary*). Exemplo: o usuário deseja restringir a temperatura ambiente entre 10 e 50C, pois não gosta de sentir frio. Mesmo que o sensor tenha capacidade, a temperatura não extrapolará os limites estabelecidos pela preferência do usuário.

Os exemplos abaixo reproduzidos a partir da norma (HAN; YOON, 2012) mostram o uso da CIDL, em conjunto dos vocabulários DCDV, SCDV, SEPV e SAPV.

```
<cidl:SensoryDeviceCapability xsi:type="dcdv:ScentCapabilityType"
id="scent01" maxIntensity="5" numOfLevels="2" locator="urn:mpeg:mpeg-v:01-
SI-LocationCS-NS:center">
  <dcdv:Scent>urn:mpeg:mpeg-v:01-SI-ScentCS-NS:rose</dcdv:Scent>
</cidl:SensoryDeviceCapability>
```

Esse exemplo da CIDL com o DCDV ilustra a descrição das capacidades de um dispositivo que produz aromas. O dispositivo identificado por “scent01”, é capaz de atingir a intensidade máxima de um aroma de 5 ml por hora, com dois níveis de controle. O dispositivo está localizado no centro e é capaz de produzir somente perfumes de rosa.

```
<cidl:SensorDeviceCapability xsi:type="scdv:TemperatureSensorCapabilityType"
  id="TS001" maxValue="120" minValue="-20" numOfLevels="1400" offset="1.0"
  unit="celsius">
  <cidl:Accuracy xsi:type="cidl:ValueAccuracy" value="0.1"/>
  <scdv:Location>
    <mpegvct:X>1.0</mpegvct:X>
    <mpegvct:Y>1.0</mpegvct:Y>
    <mpegvct:Z>-1.0</mpegvct:Z>
  </scdv:Location>
</cidl:SensorDeviceCapability>
```

Esse exemplo da CIDL com o SCDV mostra a descrição das capacidades de um sensor de temperatura identificado por “TS001”, capaz de obter informações de temperatura que variam entre -20 e 120C, perfazendo 1200 níveis de temperatura. A acurácia é de 0,1C. A informação detectada é recebida na localização 1,00, 1,00, -1,00 (x, y e z).

```
<cidl:USPreference xsi:type="sepv:LightPrefType" activate="true"
```

```
unit="urn:mpeg:mpeg-v:01-CI-UnitTypeCS-NS:lux" maxIntensity="300">
<sepv:UnfavorableColor>
:mpeg:mpeg-v:01-SI-ColorCS-NS:alice_blue
</sepv:UnfavorableColor>
</cidl:USPreference>
```

Esse exemplo da CIDL com o SEPV mostra a descrição da preferência do usuário por um efeito de luz de no máximo 300 *lux*. A cor “alice_blue” não é desejada.

```
<cidl:SAPreference xsi:type="sapv:TemperatureAdaptationPrefType" id="TSAP001"
  activate="true" minValue="0" maxValue="100" sensorAdaptationMode="strict"
  numOfLevels="10"/>
```

Esse exemplo da CIDL com o SAPV mostra a descrição de preferência do usuário para adaptação dos sensores de temperatura. A temperatura desejada está entre 0 e 100C e terá somente 10 níveis entre os valores mínimo e máximo. Quando os valores da preferência não estão dentro do intervalo das capacidades dos dispositivos, eles devem ser ajustados adequadamente pelo motor de adaptação.

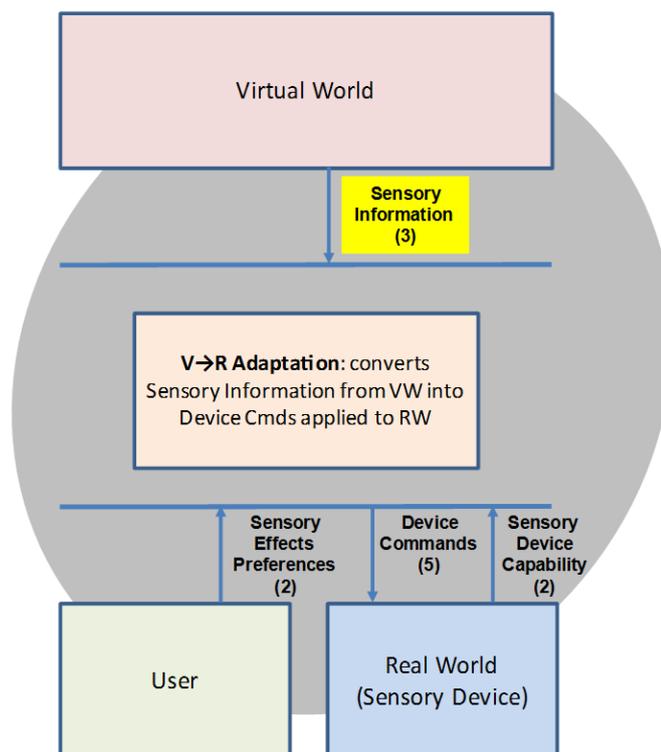
2.3 MPEG-V Parte 3 – Efeitos Sensoriais

A parte 3 do MPEG-V (ISO/IEC 23005-3) estrutura a descrição de informações sensoriais para estimular outros sentidos além da visão ou audição, como por exemplo, termocetores, mecanocetores e quimiocetores (CHOI; KIM, 2012). Ela especifica a sintaxe e a semântica para descrever informações sensoriais sendo aplicável para melhorar a qualidade da experiência dos usuários enquanto consumidores de conteúdo multimídia. As informações sensoriais podem ser encaminhadas diretamente para os dispositivos atuadores no mundo real ou então sofrerem uma transformação no motor de adaptação (não normativo) gerando como saída comandos mais simples para dispositivos compatíveis com a parte 5 do MPEG-V. A parte 3 da norma interage com as partes 2 e 5 da ISO/IEC 23005. A arquitetura e seu escopo (destacado em amarelo) são apresentados na Figura 5.

A parte 3 do MPEG-V especifica uma linguagem baseada em XML para descrição de efeitos sensoriais (SEDL - *Sensory Effect Description Language*), tais como luminosidade, aroma, vento, névoa, etc (CHOI; KIM, 2012). Os tipos de efeitos sensoriais não são parte da linguagem SEDL, mas são definidos em um vocabulário de efeitos sensoriais (SEV - *Sensory Effect Vocabulary*), fornecendo extensibilidade e flexibilidade para que desenvolvedores definam seus próprios efeitos sensoriais (KIM; HAN, 2014). Uma descrição de metadados de efeitos sensoriais (SEM - *Sensory Effect Metadata*) pode estar associada a qualquer tipo de conteúdo multimídia, seja ele um jogo, um filme, uma música ou um website.

Waltl, Timmerer e Hellwagner (2009) conceituam um cenário que forma a cadeia

Figura 5 – Escopo da parte 3 do MPEG-V destacada na caixa amarela e sua arquitetura.



Fonte: (CHOI; KIM, 2012)

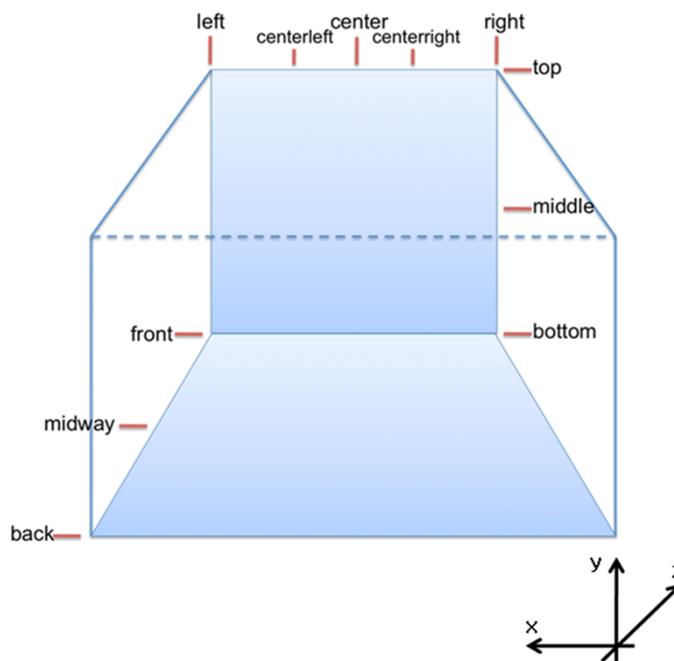
da entrega das mídias até a renderização nos tocadores e atuadores. A mídia e o SEM podem ser obtidos a partir de uma fonte tradicional como DVD ou *Blu-Ray*, ou ainda, de serviços online. Em seguida o motor de adaptação faz a mediação para interpretar os recursos de mídia acompanhados dos efeitos sensoriais de forma sincronizada, baseado nas preferências dos usuários e nas capacidades dos dispositivos. Assim, todo o conteúdo produzido pode ser entregue de acordo com o cenário específico de cada usuário para a renderização.

É importante ressaltar que não há necessariamente um mapeamento um-para-um (1:1) entre elementos ou tipos de dados de efeitos e os dispositivos sensoriais (KIM; HAN, 2014). Um equipamento com capacidade de gerar temperaturas quente e fria poderia atender o propósito de 2 tipos de efeitos sensoriais: calor e frio. Além desses 2 tipos, atualmente a norma define metadados (SEV) para efeitos de luz, luz colorida, *flash* de luz, vento, vibração, vaporização, perfume, névoa, correção de cor, movimento, cinestesia e tato (CHOI; KIM, 2012).

Na descrição de um efeito sensorial é possível orquestrar a renderização dos efeitos em diversos dispositivos localizados em posições distintas no ambiente real do usuário. Nessa direção, um modelo de localização para os metadados de efeitos sensoriais (Figura 6) é definido no esquema de classificação “LocationCS” na ISO/IEC 23005-6 (YOON; HAN,

2012) e usado para apoiar a parte 3 do MPEG-V.

Figura 6 – Modelo de localização para metadados de efeitos sensoriais e sistema de coordenadas de referência.



Fonte: (CHOI; KIM, 2012)

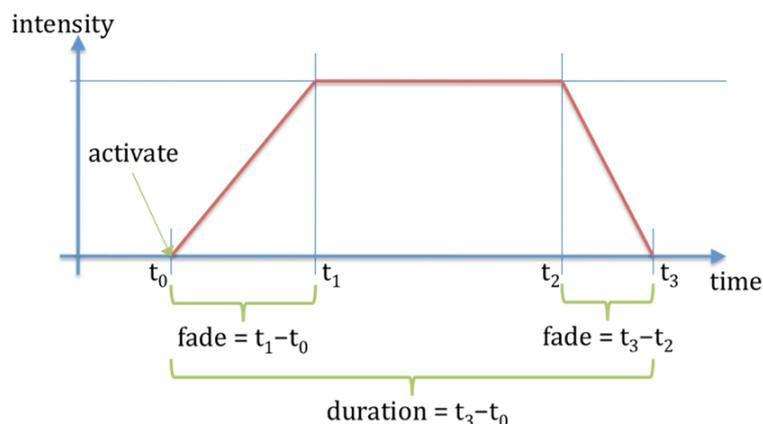
Os termos definidos no esquema “LocationCS” devem ser concatenados com o caractere “:” seguido da ordem do eixo x, y, e z para definir um local dentro do espaço tridimensional. *Wild cards* com o sinal “*” podem ser usados para se referir a um grupo de localização. Por exemplo, o valor “urn:mpeg:mpeg-v:01-SI-LocationCS-NS:*:*:back” indica que o efeito será renderizado considerando os atuadores que estejam em qualquer posição de x e y mas o eixo z restringe o efeito à parte de trás (*back*) do usuário.

A norma define também um modelo temporal para apoiar a descrição de metadados de efeitos sensoriais (CHOI; KIM, 2012). Através do modelo é possível aumentar ou diminuir gradativamente a intensidade de um efeito sensorial, conforme mostra a Figura 7.

Para a compreensão do modelo, suponha um efeito de vento ativado em t_0 e desativado em t_3 . Sua duração é t_3-t_0 e ele pode conter uma descrição para um aumento gradual, que é atribuído com duração de t_1-t_0 . A duração t_1-t_0 é o tempo dentro do qual o efeito de vento deve atingir sua intensidade especificada. Da mesma maneira, uma diminuição gradual ocorre em t_2-t_3 . Os trechos com descrições em XML apresentados no final nessa seção ilustram um exemplo prático desse tipo de cenário.

Com o intuito de abstrair e facilitar o entendimento da SEDL, Waltl, Timmerer e Hellwagner (2009) apresentaram uma descrição formal da mesma por meio do Formalismo EBNF (*Extended Backus-Naur Form*). A seguir a descrição é apresentada e conceituada.

Figura 7 – Modelo de tempo para metadados de efeitos sensoriais.



Fonte: (CHOI; KIM, 2012)

```

SEM ::= [DescriptionMetadata]
      (Declarations|GroupOfEffects|Effect|ReferenceEffect)+

Declarations ::= (GroupOfEffects|Effect|Parameter)+

GroupOfEffects ::= timestamp
                EffectDefinition
                EffectDefinition (EffectDefinition)*

Effect ::= timestamp
        EffectDefinition

EffectDefinition ::= [activate] [duration] [fade-in] [fade-out]
                   [alt] [priority] [intensity] [position] [adaptability]

```

O SEM é o elemento raiz e pode conter opcionalmente a descrição de metadados (*DescriptionMetadata*) que provê informações sobre o SEM. Em seguida, pelo menos uma das opções entre *Declarations*, *GroupOfEffects*, *Effect*, e *ReferenceEffect* deve ser fornecida. O elemento *Declarations* é usado para definir configurações comuns usadas pelos vários efeitos sensoriais e deve conter pelo menos uma das opções entre *GroupOfEffects*, *Effect* e *Parameter*. O elemento *GroupOfEffects* começa com uma marca de tempo (*timestamp*) que fornece quando o grupo de efeitos estará disponível para a aplicação e pode ser usada para fins de sincronização com a mídia associada. Também deve conter pelo menos duas definições de efeitos (*EffectDefinitions*) que correspondem a informações de efeitos sensoriais. O elemento *ReferenceEffect* permite referenciar um efeito criado anteriormente ou um grupo de efeitos. O elemento *Effect* é usado para descrever um efeito sensorial em uma determinada marca de tempo (*timestamp*). O elemento *EffectDefinition* possui atributos opcionais para configuração do efeito. *Activate* descreve se o efeito deve ser ativado; *duration*

define a duração do efeito; *fade-in* e *fade-out* correspondem ao aumento/diminuição gradual do efeito; *alt* descreve um efeito alterativo caso o original não possa ser processado; *priority* diz respeito a prioridade entre os efeitos de um mesmo grupo; *intensity* define a intensidade do efeito de acordo com uma escala; *position* descreve a posição onde será renderizado o efeito em relação ao usuário, e por fim; *adaptability* que define tipos preferidos de adaptação do efeito correspondente.

Os exemplos abaixo reproduzidos a partir da norma (CHOI; KIM, 2012) mostram o uso da SEDL em conjunto do vocabulário SEV, apresentando sequencialmente os tipos de efeitos para luz, vento e vibração.

```
<sedl:Effect xsi:type="sev:LightType" intensity-value="50.0" intensity-range="0.00001
  32000.0" activate="true" color="#FF0000" si:pts="0"/>
...
<sedl:Effect xsi:type="sev:LightType" activate="false" color="#FF0000" si:pts="28"/>
```

Esse exemplo da SEDL com o SEV mostra a descrição de um efeito de luz. A intensidade na ativação é de 50 *lux* (em um intervalo entre 0.00001 e 32000) com a cor “#FF0000”. O efeito é ativado em `si:pts="0"` e desativa em `si:pts="28"`.

```
<sedl:DescriptionMetadata>
  <sedl:ClassificationSchemeAlias alias="WCS"
    href="urn:mpeg:mpeg-v:01-SI-LocationCS-NS"/>
</sedl:DescriptionMetadata>
<sedl:Effect xsi:type="sev:WindType" fade="5" location=":WCS:*:*:front"
  intensity-value="3.0" intensity-range="0.0 12.0" activate="true" si:pts="0"/>
```

Esse exemplo da SEDL com o SEV mostra a descrição de um efeito de vento. O efeito é ativado em `si:pts="0"`, com a intensidade em 3.0 (em um intervalo entre 0 e 12 de acordo com a escala *Beaufort*) atingida após 5 segundos de aumento gradativo, para qualquer posição de x e y do atuador que esteja na frente do usuário.

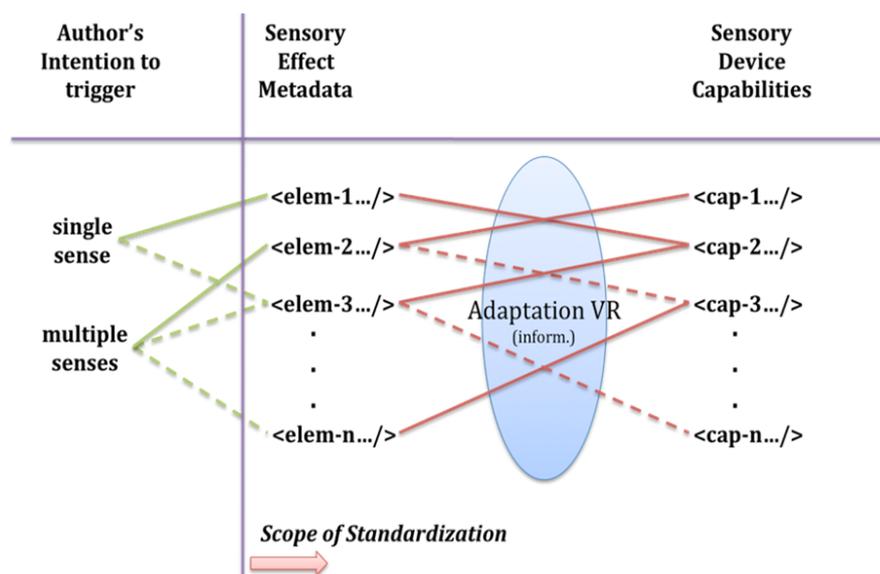
```
<sedl:Effect xsi:type="sev:VibrationType" intensity-value="4.1" intensity-range="0.0
  50.0" duration="7" fade="3" si:pts="0"/>
```

Esse exemplo da SEDL com o SEV mostra a descrição de um efeito de vibração. O efeito é ativado em `si:pts="0"` com intensidade de 4.1 (em um intervalo entre 0 e 50 de acordo com a escala *Hertz*) com duração de 7 segundos e reduzido gradativamente a 3 segundos do fim.

O propósito básico da ISO/IEC 23005-3 é apresentado na Figura 8 por um mapeamento das intenções do autor de conteúdo para metadados de efeitos sensoriais e

capacidades de dispositivos (CHOI; KIM, 2012). Em linhas gerais, um autor de conteúdo pretende estimular um sentido único ou múltiplos sentidos a partir da anotação de recursos multimídia com os metadados de efeitos sensoriais. Uma ferramenta de criação pode fornecer esse mapeamento e como saída gerar SEM compatível com a especificação da parte 3 do MPEG-V.

Figura 8 – Mapeamento das intenções do autor dos metadados de efeitos sensoriais com a capacidade dos dispositivos.



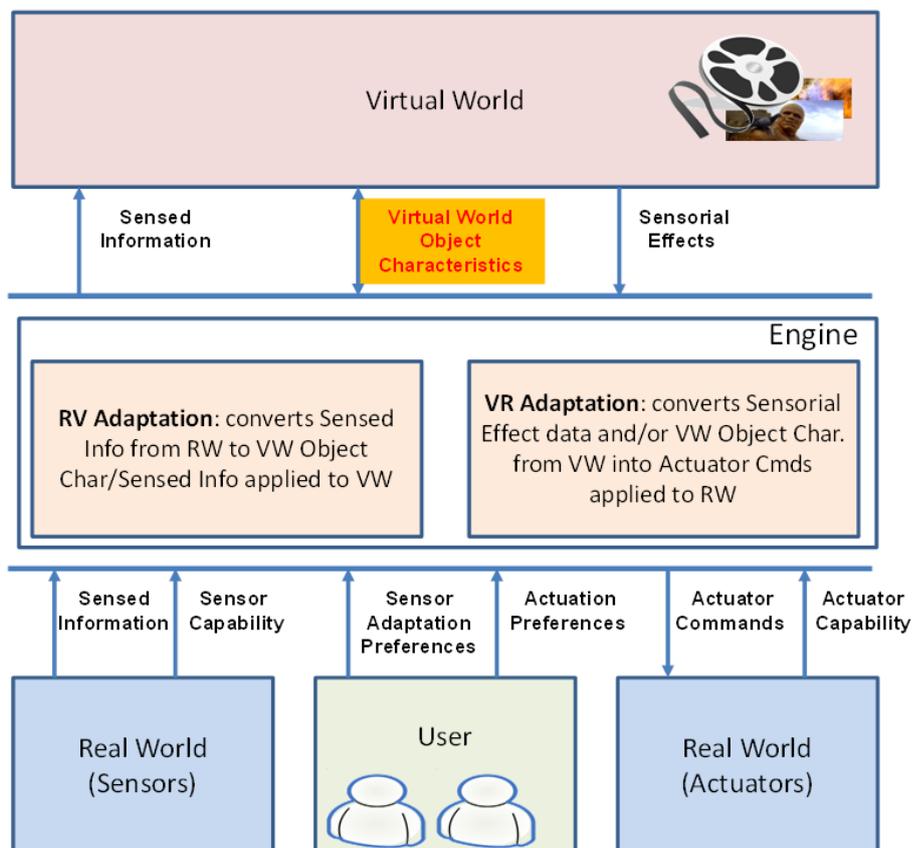
Fonte: (CHOI; KIM, 2012)

2.4 MPEG-V Parte 4 – Características de Objetos do Mundo Virtual

A parte 4 do MPEG-V (ISO/IEC 23005-4) especifica sintaxe e semântica para descrever características de objetos do mundo virtual para torná-lo compatível com outro mundo virtual ou permitir que um objeto virtual seja controlado a partir de entradas (*inputs*) do mundo real (HAN; PREDA, 2012). Seu escopo pode ser observado na Figura 9.

Em um processo de *design* de ambientes virtuais, a criação de objetos pode ser uma tarefa demorada, pois envolve a caracterização individual de cada um associada a comportamentos, eventos e outras propriedades. A ideia principal do padrão é permitir a criação do objeto uma única vez e que ele possa ser utilizado em diferentes ambientes virtuais e que também possa ser compatível com propriedades do mundo real, por exemplo, informações de sensores fisiológicos dos usuários poderiam ser transportadas para o ambiente virtual com a utilização dos metadados descritos no padrão (KIM; HAN, 2014).

Figura 9 – Escopo da parte 4 do MPEG-V destacada na caixa amarela.



Fonte: (KIM; HAN, 2014)

O tipo de base das características de objetos do mundo virtual é composto pelos seguintes tipos de dados (HAN; PREDA, 2012):

- Identidade: contém um descritor de identificação como usuário associado ao objeto virtual, proprietário, direitos, etc.
- Som: contém recursos de som e propriedades relacionadas.
- Perfume: contém recursos de cheiro e as propriedades relacionadas.
- Controle: contém um conjunto de descritores para controlar movimento de um objeto virtual, orientação e escala.
- Evento: contém um conjunto de descritores que proporcionam eventos de entrada a partir de um mouse, teclado, etc.
- Modelo de comportamento: contém um conjunto de descritores que definem as informações comportamento do objeto de acordo com os eventos de entrada.
- Id: contém um valor exclusivo para identificar cada objeto virtual individualmente.

Para descrever os metadados de avatares e de objetos virtuais, o tipo base é herdado e aspectos específicos são usados para cada metadado respectivo. Os metadados de um avatar no ambiente virtual podem ser utilizados para representar o usuário real no ambiente virtual e permitir a interação com o ambiente virtual. Os metadados são compostos de tipos de dados que descrevem recursos de aparência, recursos de animação, habilidades de comunicação, personalidade, recursos de controle, propriedades táteis e de gênero. Os metadados de um objeto virtual têm o propósito de caracterizar vários tipos de objetos no ambiente virtual e oferecer uma forma de interação com cada objeto. Os metadados descrevem os tipos de dados específicos e permitem descrever recursos de aparência, recursos de animação, propriedades táteis e componentes de objetos virtuais (KIM; HAN, 2014).

Um trecho da norma ISO/IEC 23005-4 (HAN; PREDÁ, 2012) contendo um exemplo de descrição de um avatar e de objetos virtuais associados ao ambiente é apresentado a seguir:

```
<vwoc:VWOCInfo xsi:schemaLocation="urn:mpeg:mpeg-v:2012:01-VWOC-NS VWOCSchema.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:mpegvct="urn:mpeg:mpeg-v:2012:01-CT-NS"
  xmlns:vwoc="urn:mpeg:mpeg-v:2012:01-VWOC-NS"
  xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004">
  <vwoc:AvatarList>
    <vwoc:Avatar xsi:type="vwoc:AvatarType" id="AVATARID_1" gender="male">
      <vwoc:VWOC>
        <vwoc:SoundList>
          <vwoc:Sound loop="1" soundID="SOUNDID_10" duration="10" intensity="3"
            name="BurpSound">
            <vwoc:ResourcesURL>http://www.BurpSound.info</vwoc:ResourcesURL>
          </vwoc:Sound>
        </vwoc:SoundList>
        <vwoc:ScentList>
          <vwoc:Scent loop="2" duration="1" intensity="3" name="BurpingScent"
            scentID="SCENTID_11">
            <vwoc:ResourcesURL>http://www.Burp.info</vwoc:ResourcesURL>
          </vwoc:Scent>
        </vwoc:ScentList>
        <vwoc:ControllList>
          <vwoc:Control controlID="CTRLID_12">
            <vwoc:MotionFeatureControl>
              <vwoc:Position>
                <mpegvct:X>1</mpegvct:X>
                <mpegvct:Y>1</mpegvct:Y>
                <mpegvct:Z>10</mpegvct:Z>
              </vwoc:Position>
            </vwoc:MotionFeatureControl>
          </vwoc:Control>
        </vwoc:ControllList>
      </vwoc:VWOC>
    </vwoc:Avatar>
  </vwoc:AvatarList>
</vwoc:VWOCInfo>
```

```

    <vwoc:Orientation>
      <mpegvct:X>0</mpegvct:X>
      <mpegvct:Y>0</mpegvct:Y>
      <mpegvct:Z>0</mpegvct:Z>
    </vwoc:Orientation>
    <vwoc:ScaleFactor>
      <mpegvct:X>1</mpegvct:X>
      <mpegvct:Y>1</mpegvct:Y>
      <mpegvct:Z>3</mpegvct:Z>
    </vwoc:ScaleFactor>
  </vwoc:MotionFeatureControl>
</vwoc:Control>
</vwoc:ControlList>
<vwoc:EventList>
  <vwoc:Event eventID="ID_13">
    <vwoc:Mouse>urn:mpeg:mpeg-v:01-VWOC-MouseEventCS-NS:click
    </vwoc:Mouse>
  </vwoc:Event>
</vwoc:EventList>
</vwoc:VWOC>
<vwoc:BehaviorModelList>
  <vwoc:BehaviorModel>
    <vwoc:BehaviorInput eventIDRef="ID_13"/>
    <vwoc:BehaviorOutput controlIDRefs="CTRLID_12" scentIDRefs="SCENTID_11"
      soundIDRefs="SOUNDID_10"/>
  </vwoc:BehaviorModel>
</vwoc:BehaviorModelList>
</vwoc:Avatar>
</vwoc:AvatarList>
</vwoc:VWOCInfo>

```

Explicando o exemplo, há um avatar com id="AVATARID_1", do gênero masculino, que possui um recurso de som, cheiro e controle de movimento associado. A referência de associação pode ser para um recurso interno ou externo. Há um controle de movimento identificado como "CTRLID_12" e um evento para *click* do mouse com id "ID_13". Uma lista associa comportamentos de entrada com as saídas desejadas. Quando ocorrer o evento *click* do mouse, o avatar executará as animações descritas nas propriedades "CTRLID_12" (controle de movimento), "SCENTID_11" (cheiro) e "SOUNDID_10" (som).

O MPEG-V parte 4 foi concebido para representar as características de objetos do mundo virtual e não inclui informação de geometria, som, animação, textura. Entretanto, o MPEG-V parte 4 pode ser combinado com o MPEG-4 Parte 11 (elementos gráficos) e Parte 16 (definir e animar avatares) em um ambiente interativo (HAN; PREDA, 2012).

O anexo da norma traz ainda descrições de esquemas de classificação que são utilizados para compor características de objetos virtuais, como por exemplo, esquemas de

classificação para eventos de mouse, animações, etc.

2.5 MPEG-V Parte 5 – Formato de Dados para Dispositivos de Interação

A parte 5 do MPEG-V (ISO/IEC 23005-5) tem como objetivo fornecer formato de dados para dispositivos de interação que podem ser sensores ou atuadores. Não apenas o padrão MPEG-V pode ser beneficiado dessa norma, outros padrões internacionais como ISO/IEC 23007-2 (MPEG-U) ou ISO/IEC 14496-20 (especificações de representações de cenas) se referem ao uso do formato definido no padrão (KIM; HAN; YOON, 2012).

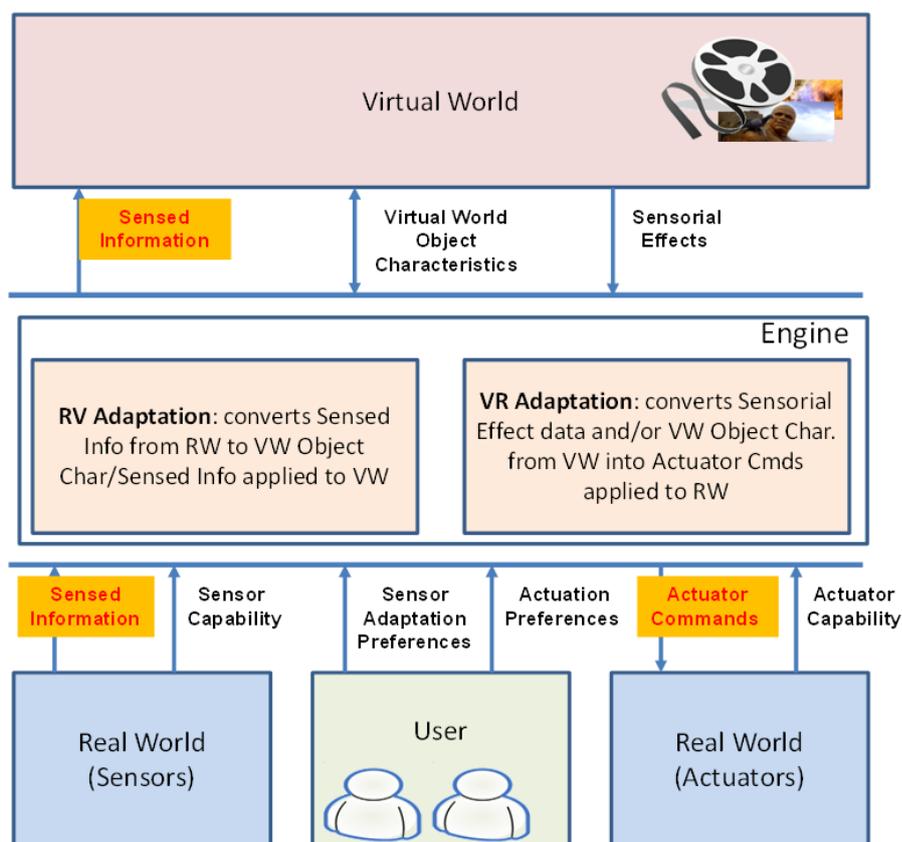
Na norma, é especificada sintaxe e semântica para formato de dados para dispositivos de interação, que são descrições de comandos para atuadores (Mundo Virtual para Real) e descrições de informações obtidas a partir de sensores no mundo real (Mundo Real para Virtual). Seu escopo pode ser observado na Figura 10. Uma linguagem para descrição de interface para interação (IIDL - *Interaction Interface Description Language*) baseada em XML permite definir interfaces para a troca de mensagens entre o mundo real e virtual e vice-versa (KIM; HAN; YOON, 2012).

Como parte da IIDL, dois vocabulários apoiam a descrição das mensagens para enviar os comandos (DCV - *Device Command Vocabulary*) ou obter informações a partir dos sensores (SIV - *Sensed Information Vocabulary*). O vocabulário DCV define descrições para comandar dispositivos atuadores de maneira mais simples que o vocabulário de efeitos sensoriais (SEV), por ser curta (KIM; HAN, 2014). No entanto, para reproduzir um efeito sensorial de incremento de vento, por exemplo, mais de um comando com intensidades diferentes precisam ser descritos sequencialmente. Pode ser mais razoável para um fabricante implementar um atuador compatível com MPEG-V parte 5 pela simplicidade na execução e num cenário de efeitos sensoriais descritos com MPEG-V parte 3, pode ser necessário um motor de adaptação, conforme apresentado na Figura 3. O vocabulário SIV define descrições para obter informações a partir de sensores do mundo real, tais como luz, temperatura, acelerômetros, bio sensores, etc (KIM; HAN, 2014).

Os exemplos abaixo reproduzidos a partir da norma (KIM; HAN; YOON, 2012) mostram o uso da IIDL em conjunto dos vocabulários DCV e SIV:

```
<iidl:InteractionInfo>
  <iidl:DeviceCommandList>
    <iidl:DeviceCommand xsi:type="dcv:WindType" id="wind01"
      deviceIdRef="wind001" activate="true" intensity="30">
      <iidl:TimeStamp xsi:type="mpegvct:AbsoluteTimeType" absTime="1:30:23"/>
    </iidl:DeviceCommand>
```

Figura 10 – Escopo da parte 5 do MPEG-V destacada em caixas amarelas.



Fonte: (KIM; HAN, 2014)

```
</iidl:DeviceCommandList>
</iidl:InteractionInfo>
```

Esse exemplo da IIDL com o DCV mostra um comando de vento para um dispositivo que dispara vento. O comando identificado por “wind01” para o dispositivo identificado com o id “wind001”, com intensidade de 30% da intensidade máxima no tempo absoluto decorrido “1:30:23”.

```
<iidl:InteractionInfo>
  <iidl:SensedInfoList>
    <iidl:SensedInfo xsi:type="siv:TemperatureSensorType" id="TS001"
      sensorIdRef="TSID001" activate="true" value="36.5">
      <iidl:TimeStamp xsi:type="mpegvct:ClockTickTimeType" timeScale="100"
        pts="60000"/>
    </iidl:SensedInfo>
  </iidl:SensedInfoList>
</iidl:InteractionInfo>
```

Esse exemplo da IIDL com o SIV mostra a obtenção da informação de um sensor de

temperatura. A informação que precisa ser obtida tem o identificador “TS001” com referência para o dispositivo com id “TSID001”. O Sensor será ativado com o valor 36.5C. Deve ser mantido até o tempo “60000” e o valor colhido a cada “100” pulsos por segundo.

2.6 MPEG-V Parte 6 – Ferramentas e Tipos Comuns

A parte 6 do MPEG-V (ISO/IEC 23005-6) especifica sintaxe e semântica para ferramentas e tipos de dados comuns para uso nas outras partes do padrão, como por exemplo tipos relacionados a cores, tempo, unidades, etc.

Contém um anexo que fornece esquemas de classificação para os tipos de dados que são identificadas por uma sequência única de caracteres formada pela URN “urn:mpeg:mpeg-v:01-CI-NameCS-NS”, em que o nome (*Name*) deve ser substituído com o nome do esquema de classificação (YOON; HAN, 2012). Por exemplo, a URN “urn:mpeg:mpeg-v:01-CI-UnitTypeCS-NS” identifica o esquema de classificação que prevê valores para *UnitTypeCS*, que podem ser centímetros, metros, quilômetros, milhas, polegadas, etc.

Além do esquema de classificação *UnitTypeCS*, outros esquemas estão incluídos no anexo da norma: *ColorCS*, *LocationCS*, *ScentCS*, *ShakeDirectionCS*, *SpinDirectionCS*, *SprayingTypeCS*, *TactileEffectCS*, *WaveDirectionCS*, *WaveStartDirectionCS*, *TactileDisplayCS*, e esquemas de classificação de características do esqueleto humano.

A seguir, uma reprodução do trecho da norma (YOON; HAN, 2012) contendo 3 termos para o esquema de classificação de cores (*ColorCS*):

```
<ClassificationScheme uri="urn:mpeg:mpeg-v:01-SI-ColorCS-NS">
  <Term termID="alice_blue">
    <Name xml:lang="en">Alice blue</Name>
    <Definition>Describes the color Alice blue. Hex: #F0F8FF; RGB: 240, 248, 255; HSV:
      208 degrees, 6%, 100%.</Definition>
  </Term>
  <Term termID="alizarin">
    <Name xml:lang="en">Alizarin</Name>
    <Definition>Describes the color Alizarin. Hex: #E32636; RGB: 227, 38, 54; HSV: 355
      degrees, 83%, 89%.</Definition>
  </Term>
  <Term termID="amaranth">
    <Name xml:lang="en">Amaranth</Name>
    <Definition>Describes the color Amaranth. Hex: #E52B50; RGB: 229, 43, 80; HSV: 345
      degrees, 78%, 64%.</Definition>
  </Term>
```

O identificador do termo é acrescentado à URI do esquema de classificação para formar uma URN que representa as cores RGB descritas na definição do termo. Desse

modo, a URN que representa a cor “Alice blue”, por exemplo, é “urn:mpeg:mpeg-v:01-SI-ColorCS-NS:alice_blue”.

2.7 MPEG-V Parte 7 – Conformidade e Software de Referência

A parte 7 do MPEG-V (ISO/IEC 23005-7) especifica regras de conformidade e software de referência com a finalidade de validar e esclarecer as especificações descritas com as várias partes do MPEG-V. Os módulos de software de referência disponíveis são especificados na forma de interfaces de programação de aplicativos (API), de acordo com a norma ISO/IEC 23006 (KIM; JOO; CHOI, 2013).

Um esquema de validação baseado no Schematron (KIM; JOO; CHOI, 2013), definido na parte 3 da ISO/IEC 19747-3 (*Document Schema Definition Languages – Part 3: Rule-based validation – Schematron*), é fornecido para testes de conformidade. As regras são executadas em conjunto da ferramenta Saxon para validar as descrições.

Um trecho de validação reproduzida a partir da norma (KIM; JOO; CHOI, 2013) é descrita a seguir:

```
<pattern name="SEM element">
  <!-- R1.0: Check the conformance of SEM -->
  <rule context="sedl:SEM">
    <assert test="@si:timeScale">
      The SEM element shall have a timeScale attribute.
    </assert>
  </rule>
</pattern>
```

Se um elemento “SEM” de um arquivo de metadados de efeitos sensoriais não contém o atributo “si:timeScale”, então, a descrição não está em conformidade.

2.8 Conclusão

Nesse capítulo foi apresentado o padrão MPEG-V, cujo foco está na interoperabilidade entre o mundo real e o mundo virtual em aplicações computacionais. Foram mostrados exemplos sobre como utilizar as especificações do padrão. A parte 1 mostra a descrição da arquitetura. A parte 2, chamada de “Informações de controle”, relata norma para capacidade do dispositivo e informações de preferência do usuário. A parte 3 especifica a SEDL, uma linguagem baseada em XML para descrição de efeitos sensoriais, tais como luminosidade, aroma, vento, névoa, etc. A parte 4 especifica as características de objetos do mundo virtual como por exemplo avatares ou objetos. A parte 5 define um formato

de dados para troca de dados entre aplicações e dispositivos do mundo real. A parte 6 especifica sintaxe e semântica para ferramentas e tipos de dados comuns para uso nas outras partes do padrão, como por exemplo tipos relacionados a cores, tempo, unidades, etc. Por último, a parte 7 especifica regras de conformidade e software de referência com a finalidade de validar e esclarecer as especificações descritas com as várias partes do MPEG-V.

A plataforma PlaySEM está fundamentada principalmente na parte 3 e também aspectos de informações sobre dispositivos relatados na parte 2. Naturalmente, faz uso da parte 6, que é utilizada como apoio das outras partes. Aspectos de referência para software observados na parte 7 também são aplicadas na solução.

Diversos trabalhos usando o padrão MPEG-V tem sido relatado na literatura, principalmente à parte 3. No próximo capítulo são apresentados os trabalhos relacionados ao tema efeitos sensoriais utilizando uma abordagem interoperável. São revisadas ferramentas de autoria de metadados de efeitos sensoriais, *Media Players* com capacidade de reprodução ou simulação de efeitos sensoriais compatíveis com o MPEG-V e atuadores para renderização de efeitos sensoriais utilizados nos principais *Media Players*.

3 Trabalhos relacionados

Nesse capítulo são discutidos alguns trabalhos relacionados aos temas autoria, reprodução e renderização de aplicações com efeitos sensoriais utilizando o padrão MPEG-V. Pesquisadores estão preocupados não somente em criar um conteúdo compatível com o padrão MPEG-V (WALTL et al., 2013; KIM, 2013; CHOI; LEE; YOON, 2011). Eles também têm se interessado em como deve ser realizada a reprodução e renderização desses conteúdos nos dispositivos atuadores no ambiente do usuário (WALTL et al., 2013; CHO, 2010; KIM; JOO, 2014).

As seções seguintes apresentam e categorizam as ferramentas extraídas dos trabalhos relacionados. As ferramentas foram categorizadas nos seguintes grupos: (1) autoria de efeitos sensoriais; (2) MPs (*Media Players*) de efeitos sensoriais e (3) atuadores para renderizadores de efeitos sensoriais. Ao final desse capítulo elas são comparadas e discutidas a partir dos atributos que os diferenciam.

3.1 Ferramentas de autoria de efeitos sensoriais

Ferramentas de autoria de efeitos sensoriais buscam propiciar facilidades ao autor de conteúdo multimídia/multissensorial por meio da integração de metadados a esses conteúdos. Geralmente, as dificuldades para escrever um código são abstraídas por meio de uma interface gráfica amigável. A utilização de um padrão para codificar os efeitos sensoriais é importante para permitir aos autores uma livre escolha.

Essa revisão concentrou-se em 3 ferramentas de autoria e reprodução de conteúdo audiovisual com efeitos sensoriais compatível com o padrão MPEG-V autoria relatadas na literatura: SEVino (*Sensory Effect Video Annotation*) (WALTL et al., 2013), RoSE (*Representation of Sensory Effects*) Studio (CHOI; LEE; YOON, 2011) e SMURF (*Sensible Media aUthoRing Factory*) (KIM, 2013). A seguir elas são apresentadas e descritas.

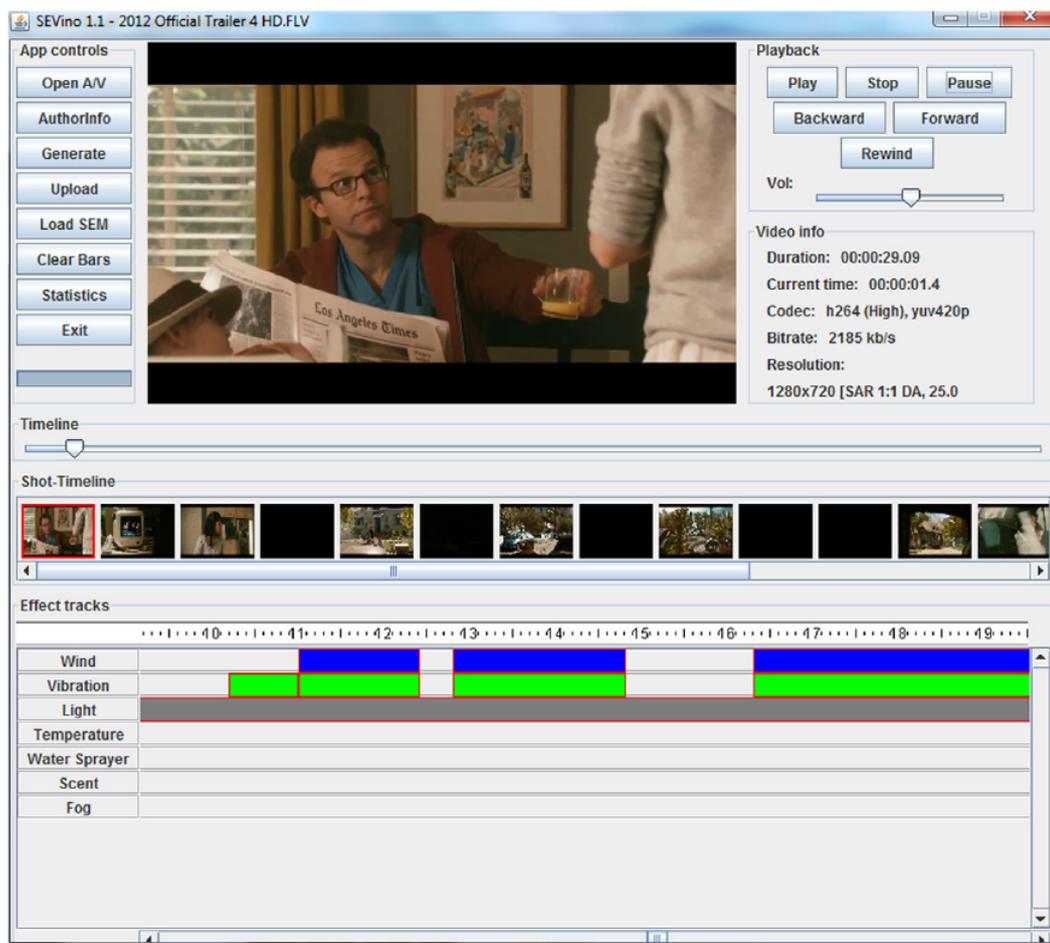
3.1.1 SEVino

Desenvolvida por Waltl et al. (2013), a ferramenta SEVino¹ permite anotar vídeos com metadados de efeitos sensoriais gerando descrições em conformidade com o padrão MPEG-V. Suporta efeitos de vento, vibração, luz, temperatura, *spray* de água, aroma e névoa. Na Figura 11 é possível visualizar sua interface de autoria.

O SEVino possui código aberto e foi desenvolvido com a linguagem Java. Há também ferramentas de terceiros integrados, como o VLCJ, para a reprodução de vídeos de

¹ SEVino: disponível em <<http://sourceforge.net/projects/sevino/>>

Figura 11 – Interface do SEVino



Fonte: (WALTL et al., 2013)

diferentes *codecs* (H.263, H.264, etc) na sua interface e o FFmpeg, que extrai quadros-chave do vídeo e disponibiliza em uma linha do tempo para facilitar a navegação do usuário.

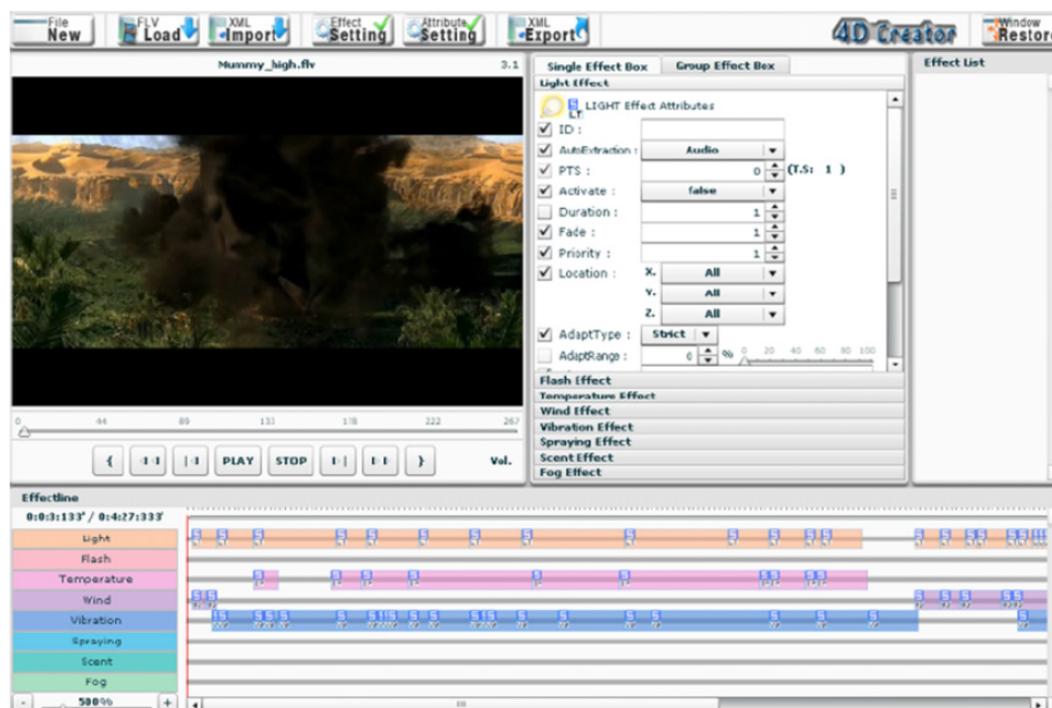
Na sua interface, informações de autoria podem ser fornecidas, sendo armazenadas em um arquivo SEM (*Sensory Effects Metadata*) no padrão MPEG-7. Há opções para controle de vídeo (retroceder, avançar, pausar, etc.), carregamento e geração de SEM, informações técnicas sobre o vídeo, estatísticas sobre a quantidade de efeitos sensoriais inseridos na anotação e edição visual do efeito na linha de tempo com definição de início e fim e as suas propriedades. Na versão 2.0 teve o software SESim integrado ao seu sistema (ver seção 3.2.2), para simulação de efeitos sensoriais.

3.1.2 SMURF

A ferramenta de autoria SMURF, acrônimo de *Sensible Media aUthoRing Factory*, foi desenvolvida por Kim (2013). Notadamente inspirada na ferramenta SEVino, a SMURF estende as funcionalidades da anterior incluindo o suporte adicional a 3 elementos do

MPEG-V: *GroupOfEffects*, *Declaration* e *ReferenceEffect* (ver capítulo 2). Sua interface foi desenvolvida com Adobe Flex, que permite construções de aplicações ricas para a Internet. O sistema de E/S é escrito em Java. A Figura 12 apresenta a interface do SMURF.

Figura 12 – Interface do SMURF



Fonte: (KIM, 2013)

O SMURF é capaz de criar efeitos sensoriais de luz, *flash* de luz, temperatura, vento, vibração, *spray* de água, aroma e névoa.

Entre suas principais funcionalidades, pode-se destacar o armazenamento e carregamento de arquivo XML contendo informações sensoriais no formato SEM, a extração automática da duração de um efeito, o registro de efeitos sensoriais, o suporte à criação de *GroupOfEffects*, *Declaration* e *ReferenceEffect* e a validação da instância XML gerada na autoria. Conforme mostra a Figura 12, sua interface é composta por 5 partes: menu de ferramentas, componente para controle de vídeo, componente para composição de efeito, componente para registrar o efeito e um componente de linha de tempo de efeitos.

3.1.3 RoSE Studio

RoSE (*Representation of Sensory Effects*) Studio é uma ferramenta de criação de conteúdo multissensorial desenvolvida por Choi, Lee e Yoon (2011). Foi pensada para um usuário-autor com dificuldades em fazer descrições de efeitos sensoriais no formato XML, que exige maior conhecimento. A Figura 13 apresenta a interface de abstração do RoSE e seus principais componentes que permitem a criação dos efeitos sensoriais.

Figura 13 – Interface do RoSE Studio e seus componentes



Fonte: (CHOI; LEE; YOON, 2011)

Sua interface é composta no lado esquerdo por um reprodutor de vídeo, que permite ao usuário navegar pelas cenas controlando por botões ou linha de tempo; no lado direito um editor de metadados de informações gerais como direitos autorais, data de criação, etc, e propriedades de efeitos sensoriais aplicados em cada definição de efeito sensorial; e na parte de baixo uma linha de tempo contendo os efeitos sensoriais adicionados. No trabalho de Choi, Lee e Yoon (2011) não são relatados os tipos de efeitos sensoriais suportados pela ferramenta, mas a partir da Figura 13, extraída do artigo, é possível ver um conjunto de ícones com representação de efeitos sensoriais em que é possível inferir alguns dos tipos de efeitos que são suportados.

Após a integração dos metadados de efeitos sensoriais para o conteúdo audiovisual, a ferramenta permite gerar o arquivo de saída de 2 formas: um arquivo XML independente contendo o SEM ou multiplexado com o conteúdo do vídeo, preparado para ser transmitido usando o padrão MPEG-2 TS (*Transport Stream*).

3.2 *Media players* de efeitos sensoriais

Para o levantamento de *Media Players* (MPs) de efeitos sensoriais, foram considerados sistemas reprodutores de vídeo capazes de simular ou reproduzir efeitos sensoriais descritos por meio do padrão MPEG-V, em outras interfaces virtuais ou físicas disponíveis no mesmo ambiente ou remoto.

Os MPs que ativam fisicamente dispositivos no mundo real fazem uso de renderizadores de efeitos sensoriais. Nesse trabalho, os renderizadores são entendidos como guias para levar a informação do mundo virtual para o mundo real e podem estar embutidos ou não nos MPs. Eles podem suportar nativamente o padrão MPEG-V ou fazer uso de APIs ou protocolos de comunicação para acionar os atuadores. As seções seguintes apresentam e descrevem alguns MPs disponíveis atualmente.

3.2.1 SEMP

O SEMP², acrônimo de *Sensory Effect Media Player*, é baseado no DirectShow³ e foi desenvolvido por Watl et al. (2013) para a reprodução de vídeos anotados com efeitos sensoriais com base no MPEG-V. Na função de renderizador de efeitos sensoriais, ele suporta os dispositivos amBX Gaming PC peripherals (luz, vento e vibração), Cyborg Gaming Lights for PC (luz) e Vortex Activ (aroma). Na Figura 14 é possível ver sua execução em conjunto do produto amBX da Phillips (relatado na seção 3.3.1).

Figura 14 – SEMP e o sistema amBX



Fonte: (WALTL et al., 2013)

O software foi desenvolvido na linguagem C++ para Windows e está disponível no formato de código aberto. Além da API do *DirectShow*, seu núcleo usa um *parser XML*, para fazer a leitura do arquivo SEM, um módulo para cálculo de cor baseado nos *frames* do vídeo com apoio do *DirectShow* (possibilitando obter 1 *frame* a cada 0,1 segundos), para auto-extração de cor, além de um gerenciador de efeitos, para controlar os dispositivos atuadores. O SEMP oferece também uma interface de votação para que o usuário possa relatar sua experiência, se for ativado.

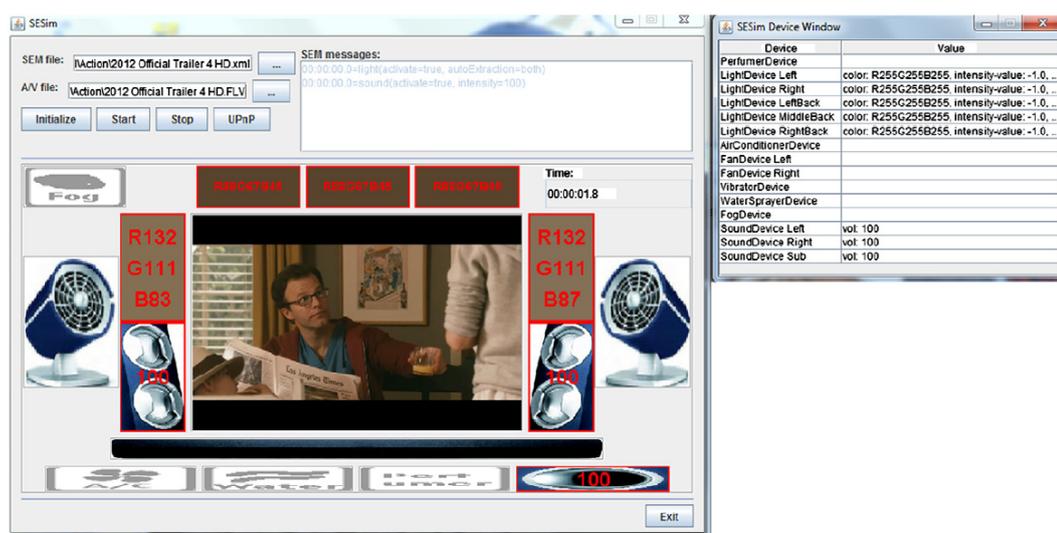
² SEMP: disponível em <<http://sourceforge.net/projects/semmediaplayer/>>

³ DirectShow – API fornecida pela Microsoft para execução de mídias no sistema operacional Windows.

3.2.2 SESim

A ferramenta SESim (*Sensory Effect Simulator*)⁴ foi desenvolvida com o intuito de simular os efeitos sensoriais criados na ferramenta de autoria SEVino (WALTL et al., 2013). Seu desenvolvimento foi separado do SEVino, apesar de atualmente elas terem sido integradas para oferecer mais comodidade ao autor de metadados de efeitos sensoriais. Sua interface apresentada na Figura 15 foi inspirada no produto ambX da Philips.

Figura 15 – Interface do SESim e sua janela com informações dos dispositivos



Fonte: (WALTL et al., 2013)

Desenvolvido em Java e disponível no formato de código aberto, sua arquitetura de componentes é muito semelhante ao SEVino: possui um *parser* XML para o arquivo SEM, usa a biblioteca VLCJ para exibir os vídeos, e conta ainda com um módulo para auto-extração automática de cores do vídeo.

A ferramenta exige como entrada um arquivo audiovisual e um arquivo SEM. Após a inicialização os dispositivos atuadores virtuais dispostos na interface são acionados de acordo com os efeitos anotados em uma linha de tempo. Para os efeitos de luz, 5 retângulos posicionados respectivamente à esquerda, centro-esquerda, centro, centro-direita e direita do vídeo são pintados de acordo com as cores extraídas dos *frames*. Os valores literais representando as cores e intensidades são escritos sob cada dispositivo. Além disso, quando acionados, os retângulos assumem uma borda na cor vermelha para indicar ao usuário que determinado dispositivo foi acionado naquele instante. A janela com informações dos dispositivos (Figura 15) também apresenta os valores assumidos por cada dispositivo em tempo real de reprodução do vídeo.

Os dados de autores do arquivo SEM não são utilizados, mas a ferramenta é capaz de analisar e verificar a conformidade do arquivo de metadados contendo essas informações.

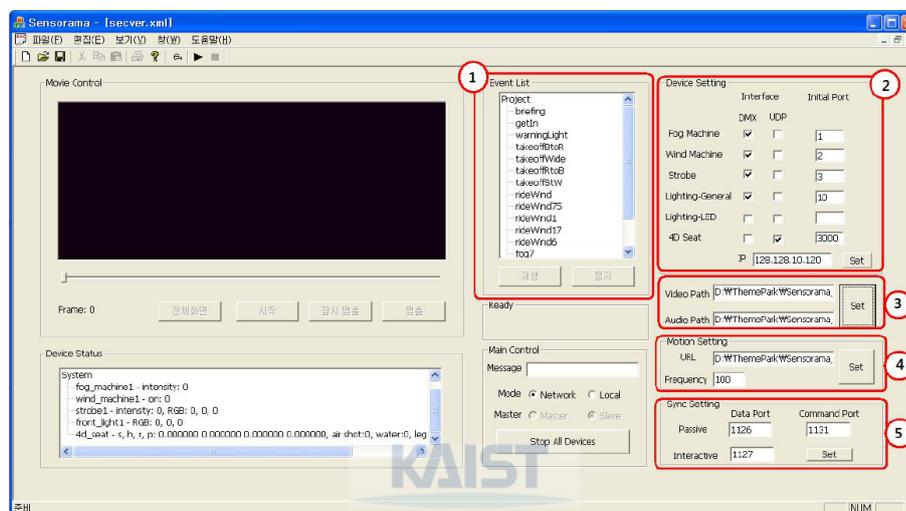
⁴ SESim: disponível em <<http://sourceforge.net/projects/sesim/>>

As descrições dos efeitos não precisam ser geradas necessariamente pelo SEVino, uma vez que o SESim é capaz de interpretar metadados de efeitos sensoriais provenientes de qualquer outra ferramenta de autoria que produza as descrições de acordo com o padrão MPEG-V, como por exemplo o RoSE Studio (WALTL et al., 2013).

3.2.3 Sensorama

A ferramenta Sensorama foi desenvolvida por Cho (2010) com a intenção de reproduzir conteúdo audiovisual com efeitos sensoriais a partir da união de metadados de efeitos sensoriais em linha de tempo com eventos sob demanda utilizando o padrão MPEG-V. Para isso, o autor descreveu blocos de efeitos sensoriais com apoio da ferramenta de autoria RoSE Studio e os enumerou em forma de uma lista de eventos na aplicação, para serem acionados a qualquer instante. Por exemplo, o efeito “explosão” pode conter uma série de efeitos de vibração, vento e calor, que podem ser reaproveitados a cada necessidade de simular uma explosão. Essa lista pode ser observada no item 1 da Figura 16.

Figura 16 – Interface do Sensorama



Fonte: (CHO, 2010)

Foi pensando na capacidade de reutilização dos efeitos e na comodidade para o manuseio de dados de efeitos para cinema 4D que Cho (2010) pensou na interface do Sensorama. O software foi construído para o sistema operacional Windows utilizando a IDE Visual C++ 2003. Os atuadores suportados são equipamentos profissionais parte de um CAVE System com suporte a vento, luz, névoa, *flashes* de luz e vibração. Na parte esquerda da Figura 16 pode ser visto o reproduzidor de conteúdo audiovisual e controle por meio de linha de tempo da ferramenta. No centro contém a lista de eventos pré-definidos, com suporte a carregamento e validação de SEM. Na direita um conjunto de configurações

para comunicação com os dispositivos atuadores. A ferramenta possui ainda suporte para sincronização de vídeo com outros dispositivos projetores.

3.2.4 Sensible Media Simulator

A ideia principal do sistema *Sensible Media Simulator* é servir de simulador de efeitos sensoriais para automóveis compatível com o padrão MPEG-V. Foi desenvolvido por Kim e Joo (2014), baseado no pensamento de aproveitar a infraestrutura de um carro moderno, composto por dispositivos de efeitos sensoriais e sensores, para levar entretenimento para automóveis. Seguindo a linha de desenvolvimento da ferramenta de autoria SMURF (KIM, 2013), ele compartilha da mesma tecnologia, Adobe Flex para interface, que permite sua utilização na web, e Java, para E/S de arquivos. A Figura 17 apresenta sua interface.

Figura 17 – Interface do Sensible Media Simulator



Fonte: (KIM; JOO, 2014)

O núcleo do sistema é o componente chamado de SMCS (*Sensible Media Control System*), responsável por controlar os efeitos sensoriais. Ele é composto por uma unidade de controle eletrônico (ECU), que controla os dispositivos do carro; um motor de renderização (SMRE - *Sensible Media Rendering Engine*), que controla o processamento dos efeitos sensoriais; um módulo para gerar e interpretar metadados XML chamado de XMM (*XML Metadata Management*); e uma unidade de controle integrado, que controla todo o sistema.

Para trocar informações entre a interface Adobe Flex e o Java, o conector BlazeDS é utilizado.

O sistema suporta os efeitos de luz, vento, temperatura e vibração, que são gerados respectivamente por meio dos seguintes dispositivos atuadores: lâmpada de LED, ventilador, ventilador com aquecimento para simular temperatura de calor e um vibrador.

3.3 Atuadores para renderizadores de efeitos sensoriais

Através de dispositivos físicos os efeitos são materializados no mundo real. Embutido ou separado do sistema operacional dos MPs, os renderizadores são responsáveis por transformar metadados de efeitos sensoriais em comandos para diversos dispositivos atuadores, como por exemplo, ventiladores, luzes, motores de vibração, etc.

São destacados nessa seção os atuadores mencionados nos trabalhos das duas seções anteriores para reprodução de efeitos sensoriais no mundo real, exceto os dispositivos domésticos mencionados por [Kim e Joo \(2014\)](#), que não puderam ser explorados por falta de informação disponível.

Cabe ressaltar que há vários outros atuadores comerciais que proporcionam imersão por meio de efeitos sensoriais, como por exemplo o Interactive Gaming Chair⁵, uma cadeira de vibração, o Game Skunk⁶, um aromatizador programável, dentre outros. Porém, não serão discutidos aqui por não serem suportados pelos principais MPs de efeitos sensoriais.

A seguir eles são descritos e caracterizados.

3.3.1 amBX Gaming PC peripherals

O produto amBX Gaming PC peripherals da [Philips \(2015\)](#) é o mais mencionado nos trabalhos que envolvem reprodução de efeitos sensoriais ([WALTL et al., 2013](#); [CHO, 2010](#); [KIM; JOO, 2014](#)). A Figura 18 apresenta o conjunto de dispositivos do sistema.

Ele é comercializado em 3 formatos diferentes, que variam de preço de acordo com as funcionalidades oferecidas: *Starter Kit*, *Pro Gamer Kit* e *Premium Kit*, o mais caro e completo. Ainda há o *Extension Kit*, que permite transformar vento e vibração nos *Kits Starter* ou *Pro Gamer*.

A tecnologia amBX oferece uma ampla experiência sensorial, já que conta com dispositivo que permite aos criadores de conteúdo adicionar efeitos no mundo real usando luz, vibração e vento. A ideia do fabricante é proporcionar um ambiente imersivo para jogos, músicas ou filmes reproduzidos a partir do computador. Para os produtores de conteúdo multimídia a Philips oferece um kit de desenvolvimento de software (SDK) que

⁵ Interactive Gaming Chair - <<http://4gamers.net/products/interactive-gaming-gaming-chair/>>

⁶ Game Skunk - <<http://www.sensoryacumen.com/>>

Figura 18 – Philips amBX Gaming PC peripherals



Fonte: (PHILIPS, 2015)

contém uma API em C puro para controlar os dispositivos, mas seu desenvolvimento foi descontinuado. Originalmente é suportado no sistema operacional Windows e para sua utilização é necessária a instalação de *drivers*.

Para jogos que não suportam originalmente o amBX, existem MODs no formato de *plugins* para que os efeitos sejam reproduzidos. Para aqueles que não tem nenhum suporte, o amBX fornece um software no ambiente Windows para gerar efeitos de iluminação automaticamente. Para os demais, a própria Philips ressalta que a produção de efeitos automáticos a partir do conteúdo não é uma tarefa trivial.

Waltl et al. (2013) se inspiraram nesse conjunto para criar o simulador SESim e embutiram suporte à tecnologia amBX no reprodutor de efeitos sensoriais SEMP.

3.3.2 Cyborg Gaming Lights for PC

O dispositivo de iluminação Cyborg Gaming Lights é fornecido pela MadCatz (2015). Ele faz uso da tecnologia amBX, originalmente criado pela Philips e atualmente é suportado pela empresa amBX UK Limited, com foco em iluminação dinâmica de ambientes. A Figura 19 apresenta seus 2 projetores de iluminação, cada um podendo gerar 16 milhões de cores.

Por fazer uso da tecnologia amBX, o procedimento para que autores de conteúdo multimídia incrementem efeitos sensoriais a jogos, músicas ou filmes é exatamente o mesmo do produto amBX Gaming PC peripherals. É suportado oficialmente somente no sistema operacional Windows. Assim como o amBX Gaming PC peripherals, ele é suportado pelo reprodutor de efeitos sensoriais SEMP (WALTL et al., 2013).

Figura 19 – Mad Catz Cyborg Gaming Lights



Fonte: (MADCATZ, 2015)

3.3.3 Vortex Activ

O Vortex Activ é um emissor de aromas programável desenvolvido pela fabricante DaleAir (2015). Os aromas são produzidos a partir de cartuchos inseridos no dispositivo, conforme mostra o kit da Figura 20. Possui interface USB e seus *drivers* foram construídos para o sistema operacional Windows.

Figura 20 – Dale Air Vortex Activ



Fonte: (DALEAIR, 2015)

O dispositivo tem a capacidade de emitir até quatro aromas de uma só vez, o que permite uma combinação de aromas em uma apresentação multissensorial. Cada cartucho contém um aroma específico e pode ser removido ou substituído quando necessário. Os aromas são emitidos por meio de ventoinhas instaladas no Vortex Activ.

Em conjunto do Cyborg Gaming Lights for PC e o amBX Gaming PC peripherals, o Vortex Activ completa a cadeia de dispositivos suportados pelo reproduzidor de efeitos sensoriais SEMP (WALTL et al., 2013).

3.3.4 CAVE System

Um sistema CAVE, acrônimo de *Cave Automatic Virtual Environment*, é apresentado por [Cho \(2010\)](#) em seu trabalho como um conjunto de atuadores usados em parques temáticos de alta tecnologia. São equipamentos profissionais de diferentes fabricantes capazes de prover vento, luz, névoa, *flashes* de luz e vibração. A Figura 21 apresenta os equipamentos usados no experimento de [Cho \(2010\)](#).

Figura 21 – CAVE System



Fonte: ([CHO, 2010](#))

O CAVE *System* é composto dos seguintes equipamentos atuadores:

- Antari Z-1200 Fog machine
 - Máquina para produzir névoa.
- CITC Hurricane II Wind machine
 - Produz vento, neblina e névoa em 3 velocidades, atingindo o máximo em 3 segundos;
 - Gira em 360 graus e inclinação de 300 graus.
- Martin Atomic 3000 Strobe
 - *Flashes* de luz ultrabilhante com lâmpada Xenon.
- Omni System SL 1040 LED Lighting
 - Iluminação do ambiente com 304 LEDs (160 vermelhos, 72 verdes e 72 azuis).
- Simuline 4DOF 4D motion chair for theaters

- Prover efeito de vibração por meio de cadeiras;
- Possui 4 eixos de movimentação.

3.4 Conclusão

Esse capítulo apresentou e categorizou várias ferramentas computacionais de apoio ao processo de produção, reprodução e apresentação de conteúdos multissensoriais em conformidade com o padrão MPEG-V. Também foram apresentados modelos atuais de dispositivos atuadores para produzir efeitos sensoriais no mundo real. Esses dispositivos funcionam como uma interface de saída para as aplicações multissensoriais descritas em MPEG-V. Nessa seção, é realizada uma comparação de aspectos das ferramentas e dispositivos atuadores citados ao longo do capítulo considerados relevantes para essa dissertação de mestrado.

Para facilitar e apoiar a comparação entre as ferramentas descritas e a proposta desse trabalho, o Quadro 1 foi criado. Na tabela, os seguintes aspectos são indicados:

- Objetivo: descreve o objetivo principal do trabalho;
- Ferramenta de autoria: indica o nome da ferramenta de autoria de metadados de efeitos sensoriais;
- MP SE: indica o nome do reprodutor de conteúdo audiovisual com suporte a metadados de efeitos sensoriais compatível com MPEG-V;
- Comunicação MP e SER (*Sensory Effects Renderer*): relaciona protocolos para comunicação entre o reprodutor de conteúdo audiovisual e o renderizador de efeitos sensoriais (mundo virtual x mundo real);
- SER independente: indica se o renderizador de efeitos sensoriais é independente do reprodutor de conteúdo audiovisual, permitindo sua reutilização com outros softwares;
- Atuadores: relaciona os hardwares que promovem os efeitos sensoriais no mundo real;
- Linguagem: indica a(s) linguagem(ns) de programação utilizadas para desenvolver a ferramenta. Uma linguagem que forneça portabilidade permite compilar um programa sobre qualquer plataforma e executá-lo no mesmo sistema ou em qualquer outro;
- Código aberto: indica se o código fonte da solução é disponibilizado, que proporciona incentivo ao desenvolvimento tecnológico;
- Multi-plataforma: capacidade dos sistemas propostos de executarem em sistemas operacionais diversos.

Quadro 1 – Comparativo dos trabalhos relacionados compostos por ferramentas de autoria, reprodução e simulação de efeitos sensoriais compatíveis com MPEG-V.

Trabalho	Objetivo	Ferramenta de Autoria	MP SE	Comunicação MP e SER	SER independente	Atuadores	Linguagem	Código aberto	Multi-plataforma
(WALTL et al., 2013)	Apresentar uma cadeia de ferramentas abertas que fornecem uma estrutura completa para produzir e reproduzir efeitos sensoriais.	SEVino	SEMP SESim (Simulador)	API (somente SEMP)	Não	amBX, Vortex Activ, Cyborg Gaming Lights	Java, C++	Sim	Não
(CHO, 2010)	Propor um método para definir efeitos reutilizáveis usando MPEG RoSE e reproduzi-los não somente em linha de tempo mas também baseado em eventos.	(Usa RoSE Studio)	Sensorama (4D Effects device control engine)	DMX, UDP	Não	CAVE System	C++	Não	Não
(KIM, 2013)	Introduzir a ferramenta de autoria SMURF, para geração efetiva de metadados de efeitos sensoriais para mídias sensíveis.	SMURF	Não apresenta	-	-	-	Adobe Flex, Java	Não	Não
(CHOI; LEE; YOON, 2011)	Apresentar o RoSE Studio, que permite autores de conteúdo 4-D construir metadados de efeitos sensoriais usando MPEG-V.	RoSE Studio	Não apresenta	-	-	-	Não especificada	Não	Desconhecido
(KIM; JOO, 2014)	Apresentar um simulador de mídia sensível e a implementação de dispositivo sensorial para provar a eficácia do sistema proposto.	(Usa SMURF)	Sensible Media Simulator	RS-232	Não	Dispositivos domésticos, controlados por transmissor	Adobe Flex, Java	Não	Não
Proposta desse trabalho	Apresentar uma plataforma doméstica para renderização de vídeos com efeitos sensoriais por meio de dispositivos reais de baixo custo compatível com MPEG-V, desacoplado interface e processamento de efeitos sensoriais.	(Usa SEVino)	PlaySEM SE Video Player	UPnP	Sim, PlaySEM SE Renderer	Arduino (Led strip light, PC Fan, Cellphone Vibrator)	Java	Sim	Sim

Fonte: Elaborado pelo autor

As ferramentas de autoria de SEM são complementares a esse trabalho no sentido de formar uma cadeia completa para produzir e reproduzir efeitos sensoriais, assim como o trabalho de [Waltl et al. \(2013\)](#). Com a funcionalidade adicional de simular os efeitos sensoriais na própria ferramenta de autoria, o SEVino se destaca em seu grupo. Com a falta de suporte para grupos de efeitos e declarações de efeitos sensoriais para serem reutilizados por meio de referências, o SEVino perde alguns pontos em relação ao SMURF, que apresenta esse suporte. Com base no objetivo do trabalho de [Cho \(2010\)](#) de reutilizar efeitos sensoriais, é possível inferir que o RoSE Studio também não apresenta esse suporte.

O SMURF apresenta sua interface web desenvolvida com Adobe Flex com a vantagem de permitir a produção de metadados de efeitos sensoriais sem a necessidade de baixar e instalar softwares no PC. Em contraposição, o software Adobe Flash, necessário para reproduzir sua interface nos navegadores web é um software proprietário e pode não ser suportado nos diversos dispositivos capazes de acessar a web, como *tablets*, *smartphones*, TVs ou mesmo PCs com sistemas operacionais sem o suporte da Adobe.

Dentre as 3 ferramentas de autoria mencionadas na literatura, somente o SEVino possui código-fonte aberto e está disponível para download. A facilidade de acesso ao software pode ajudar na popularização da ferramenta e consequente adoção massiva.

Na categoria dos MPs, nenhum trabalho apresenta o renderizador de efeitos sensoriais independente como proposto nesse trabalho. Essa característica é importante porque permite aos desenvolvedores de MPs criar softwares nativos para diversos ambientes operacionais, como TVs, sistemas CAVE e PCs, sem ter que lidar com a complexidade e restrição operacional para acionar os atuadores. Um SER independente capaz de se comunicar com MPs por meio de um padrão pode assumir a responsabilidade de interpretar e renderizar os efeitos sensoriais descritos com o MPEG-V, provendo a capacidade de reutilização do equipamento.

O SEMP, assim como o Sensorama, tem a restrição de execução no sistema operacional Windows. Entre os trabalhos relacionados, o Sensorama (na função de renderizador) é o que apresenta a maior quantidade de atuadores preparados para a renderização de efeitos sensoriais, conectado ao sistema CAVE, mas tem seu propósito questionado pelo fato do suporte do MPEG-V para reutilização de grupos de efeitos por meio de declarações (tag *Declaration*). O simulador *Sensible Media Simulator* tem uma característica interessante de usar dispositivos domésticos como simulação de dispositivos de automóveis mas não apresenta detalhes sobre a renderização por meio desses dispositivos, dificultando sua avaliação. Por usar a mesma tecnologia da ferramenta de autoria SMURF, pode apresentar as mesmas restrições operacionais. O simulador SESim pode ser muito útil para validar as anotações de SEM evitando a necessidade da infraestrutura para acomodar dispositivos reais, uma vez que os efeitos são renderizados em sua interface virtual. Sua maior virtude pode ser também sua maior desvantagem em relação aos demais, que é a falta de suporte

a atuadores reais dependendo do propósito de uso. Foram notados também alguns *bugs*, como por exemplo falha na renderização de efeitos de luz manual, que não são exibidas na interface.

Assim como nas ferramentas de autoria, a característica de código aberto também é importante nos MPs e foi encontrada no SEMP, SESim e PlaySEM SE *Video Player*.

Com relação aos atuadores utilizados pelos renderizadores de efeitos sensoriais o amBX Gaming PC peripherals se destaca pelo pioneirismo. Seu custo atual não é um atrativo e o sistema operacional é restrito ao Windows. Em contrapartida, a Philips oferece um ambiente de desenvolvimento para estimular a adesão principalmente em jogos. Quando o desenvolvedor não se preocupa em proporcionar um ambiente mais imersivo para o usuário, o software do fabricante tenta solucionar com a extração automática de cores do vídeo transcendendo-as para o ambiente. É razoável afirmar que se o atuador fosse capaz de interpretar nativamente metadados de efeitos sensoriais descritos em um padrão aberto, como o MPEG-V, a adesão de usuários possivelmente seria maior tendo em vista a facilidade apresentada para criação de efeitos nas ferramentas de autoria.

Com o custo bem menor que o equipamento da Philips, o Cyborg Gaming Lights, se aproveita da tecnologia amBX e sua infraestrutura, ampliando em teoria o público alvo. Por outro lado, os usuários que adquiram o amBX Gaming PC peripherals podem não se interessar facilmente por esse produto, uma vez que já possuem dispositivo semelhante. Também não suporta metadados de efeitos sensoriais nativamente, sendo necessário um reproduzidor para controlá-lo.

O Vortex Activ, para a reprodução de aromas, tem como potencial fator restritivo para alcançar uma experiência próxima da realidade sua limitação de produzir somente 4 aromas simultâneos. Se o produtor de efeitos sensoriais de uma mídia precisar emitir aroma de 5 frutas exibidas em um vídeo, por exemplo, esse equipamento será parcialmente útil. Para uso doméstico, seu alto custo e restrição ao sistema operacional Windows podem também não ser um atrativo. Além disso, não suporta metadados de efeitos sensoriais nativamente, sendo necessário um reproduzidor e *drivers* para controlá-lo. Ainda assim, ao estimular mais um sentido humano, como o olfato, a experiência do usuário pode ser ampliada.

O sistema CAVE apresentado pode proporcionar ao usuário uma experiência elevada, tendo em vista a capacidade operacional de seus atuadores. Apesar disso, ela ainda é altamente restritiva ao uso doméstico tanto pelo custo quanto pela infraestrutura elétrica e espacial para acomodação dos dispositivos, sendo mais empregado para fins comercial, como cinemas 4D ou parques temáticos.

Na direção oposta aos atuadores para renderizadores de efeitos sensoriais mencionados nos trabalhos relacionados, o PlaySEM SE *Renderer* possui um conjunto de atuadores

de baixo custo sem perder a característica de oferecer níveis variáveis de intensidade dos efeitos. Os atuadores para produção de efeito de vento (ventiladores de PC) e vibração (vibradores de celular) podem ser encontrados até mesmo em lixos eletrônicos. Além disso, o PlaySEM SE *Renderer* é capaz de suportar todos os outros atuadores relacionados nos trabalhos, sendo necessária sua adaptação e atualização de seu software, o que não interfere no PlaySEM SE *Video Player*.

4 Projeto e Implementação da Plataforma PlaySEM

O capítulo anterior mostrou o estado da arte com relação às principais ferramentas interoperáveis para reprodução e renderização de efeitos sensoriais usando o padrão MPEG-V. A interoperabilidade permite que mídias de diferentes produtores de conteúdo possam ser lidas e processadas por diferentes aplicações. No entanto, o maior problema das ferramentas relatadas está na impossibilidade de reuso do renderizador de efeitos sensoriais com outras aplicações.

Outra dificuldade decorrente do modelo atual de renderização de efeitos sensoriais, baseado em linha de tempo, é a indução da entrada das descrições MPEG-V nos *Media Players* por meio de uma linha sequencial única (paradigma *timeline*), diferente dos modelos geralmente desenvolvidos para aplicações baseadas em eventos.

Essa dissertação propõe uma plataforma denominada PlaySEM, um acrônimo de Reprodução (*Play*) de Metadados de Efeitos Sensoriais (SEM – *Sensory Effect Metadata*), para a renderização de efeitos sensoriais de modo desacoplado do *Media Player* usando a comunicação UPnP e compatível com o padrão MPEG-V, considerando requisitos de reutilização, mobilidade e interoperabilidade. A solução permite que o renderizador de efeitos sensoriais bem como os atuadores possam ser reaproveitados em outras interfaces de apresentação, não se limitando a aplicações baseadas em linha de tempo. Os softwares e hardwares estão disponibilizados no formato aberto, propiciando a oportunidade de reprodução de mídias anotadas com efeitos sensoriais a um baixo custo associado.

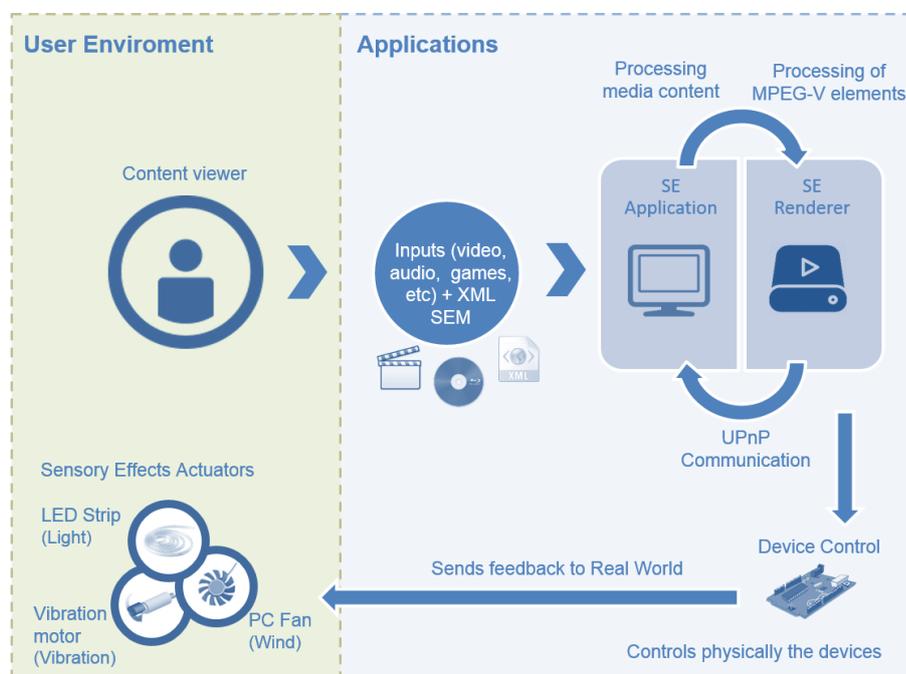
A plataforma PlaySEM é composta por três componentes:

1. PlaySEM SE *Video Player* – responsável pela reprodução de vídeo e entrada dos metadados de efeitos sensoriais;
2. PlaySEM SE *Renderer* – responsável pela transformação de metadados de efeitos sensoriais em comandos e acionamento dos dispositivos que renderizarão os efeitos sensoriais;
3. PlaySEM *Device Control* – vinculado ao SE *Renderer*, um programa embarcado em um microcontrolador que recebe os comandos para acionar os dispositivos físicos.

Em conjunto, os três softwares são capazes de reproduzir vídeos com anotações de efeitos sensoriais e renderizá-los no ambiente do usuário consumidor de multimídia. A

Figura 22 apresenta o esquema conceitual da plataforma considerando os componentes aplicados.

Figura 22 – Arquitetura conceitual da plataforma PlaySEM.



Fonte: Elaborada pelo autor

O usuário visualizador de conteúdo fornece como entrada a mídia e o SEM descrito com o padrão MPEG-V para reprodução no software *SE Video Player*, que por sua vez, se comunica com o software *SE Renderer* por meio da interface UPnP para o processamento do SEM. Nesse processo, o *SE Renderer* interpreta o SEM e gera um ou mais comandos para o *Device Control*, que contém o programa para controlar os dispositivos físicos para efeitos de vento, luz e vibração. Como retorno o usuário recebe em seu ambiente a mídia reproduzida pelo *SE Video Player* e os efeitos renderizados pelo *SE Renderer*.

A abordagem do *SE Renderer* desacoplado do *SE Video Player* para o processamento do SEM, o torna capaz de ser utilizado em qualquer *Media Player*, permitindo seu uso em dispositivos com menor poder de processamento, como por exemplo, *Smart TV's* ou *Smartphones* e *Tablets*. Esses dispositivos, por meio do protocolo Miracast, são capazes de transmitir a imagem do dispositivo para monitores compatíveis. O *SE Renderer* também tem como função disponibilizar a capacidade dos dispositivos no formato MPEG-V por meio de sua interface de serviços UPnP.

Nesse capítulo são descritos aspectos conceituais e práticos da plataforma PlaySEM, permitindo compreender suas funcionalidades, comportamento e restrições. Ao final do capítulo, uma seção de recomendações para adaptação de outros *Media Players* ao PlaySEM *SE Renderer* é apresentada.

4.1 Metodologia, ferramentas e tecnologias

Para o desenvolvimento da plataforma PlaySEM, a metodologia de análise e projeto Orientados a Objeto (OO) foi utilizada. Diagramas de Casos de uso UML foram construídos para demonstrar as possíveis interações entre atores e o sistema. Também foram desenhados diagramas comportamentais na linguagem UML: diagramas de atividades para auxiliar o entendimento do fluxo das atividades das principais operações do sistema e diagramas de sequência para apoiar a adaptação de qualquer *Media Player* ao renderizador de efeitos sensoriais (PlaySEM SE *Renderer*), demonstrando as mensagens que precisam ser trocadas e sua ordenação temporal.

No processo de projeto e desenvolvimento da plataforma, as seguintes ferramentas e linguagens foram utilizadas:

Quadro 2 – Ferramentas utilizadas no processo de desenvolvimento da plataforma PlaySEM.

Ferramenta	Linguagem	Função
Enterprise Architect 9.1.910	UML	Construção dos diagramas de casos de uso, diagramas de atividade e diagramas de sequência.
Eclipse 4.4 com o <i>plugin</i> Window-Builder	Java	Desenvolvimento do código em Java do software PlaySEM SE <i>Video Player</i> e do software PlaySEM SE <i>Renderer</i> .
Arduino IDE Sketch 1.0.6	Arduino	Desenvolvimento do software controlador dos dispositivos físicos.

Fonte: Elaborado pelo autor

Além dessas ferramentas citadas, as seguintes bibliotecas Java foram integradas aos softwares gerados a fim de auxiliar o manuseio das tecnologias utilizadas pela plataforma: (i) Cling UPnP – para comunicação UPnP; (ii) VLCJ – para reprodução de vídeos e (iii) MPEG *Metadata* – para desempacotamento de XML em classes Java.

A seguir, as tecnologias e bibliotecas mencionadas são abordadas.

4.1.1 Cling UPnP Core

A tecnologia UPnP (*Universal Plug and Play*) define uma arquitetura para conectividade pervasiva em redes para dispositivos inteligentes e computadores de todos os formatos (UPNPFORUM, 2008). A sua comunicação se baseia no protocolo HTTP e no uso da linguagem XML. Seu protocolo permite a descoberta dinâmica, descrição, controle, apresentação e envio de eventos a partir de dispositivos remotos. A ideia é permitir que dispositivos de diferentes fornecedores se comuniquem oferecendo flexibilidade e facilidade de uso.

Apesar de oferecer facilidade de uso para os usuários, trabalhar com dispositivos e serviços UPnP pode exigir uma infraestrutura não tão simples para suportar seus requisitos (CLING, 2015). A Cling UPnP Core é uma biblioteca Java que implementa essa infraestrutura, oferecendo APIs para desenvolvedores que precisam implementar serviços UPnP e pontos de controle (*control points*). Atualmente, ela é compatível com a UDA (UPnP *Device Architecture*) 1.0 e está licenciada sob a LGPL, o que permite que os arquivos JAR não modificados da biblioteca sejam utilizados em qualquer dispositivo ou aplicação para qualquer propósito.

Na arquitetura proposta a biblioteca Cling UPnP é utilizada para prover a comunicação UPnP entre o reprodutor de vídeo e os serviços do renderizador de efeitos sensoriais remoto (PlaySEM SE *Renderer*).

4.1.2 VLCJ

Um problema para reproduzir vídeos a partir da plataforma nativa do Java é a falta de *codecs* para reproduzir qualquer tipo de vídeo. O JavaFX em sua versão JRE 8, por exemplo, é uma plataforma de software multimídia que é capaz de suportar somente os formatos de vídeo FLV e MPEG-4 (ORACLE, 2015a). Apesar de serem os dois formatos mais comuns no ambiente Web, o suporte a um número mais amplo de *codecs* de vídeo é sempre interessante para plataformas multimídia.

O VLCJ é uma biblioteca *open-source* escrita na linguagem Java que permite a ligação entre aplicações Java e o reprodutor de mídia VLC *Media Player*¹ (CAPRICA, 2015). A partir dela é possível o desenvolvimento de *Media Players* em Java, abstraindo uma série de complexidades ao lidar com decodificação em diferentes formatos. Por ser baseado no VLC, também opera em multi-plataforma, fornecendo a possibilidade de escolha por parte do usuário sobre o ambiente operacional de uso. O VLC é bastante útil aos desenvolvedores uma vez que suporta praticamente todos os tipos de formatos de vídeo, sem a necessidade de instalação de *codecs* externos no sistema operacional.

Especificamente para reproduzir vídeos em qualquer formato, o VLCJ foi incorporado no software PlaySEM SE *Video Player*, apresentado na seção 4.2.2 desse capítulo.

4.1.3 MPEG Metadata

Os arquivos de metadados de efeitos sensoriais ou a descrição das capacidades do dispositivo são descritos em XML seguindo o padrão MPEG-V e precisam ser lidos pela aplicação. Para facilitar a ligação entre o XML e objetos da aplicação, foi criada a biblioteca

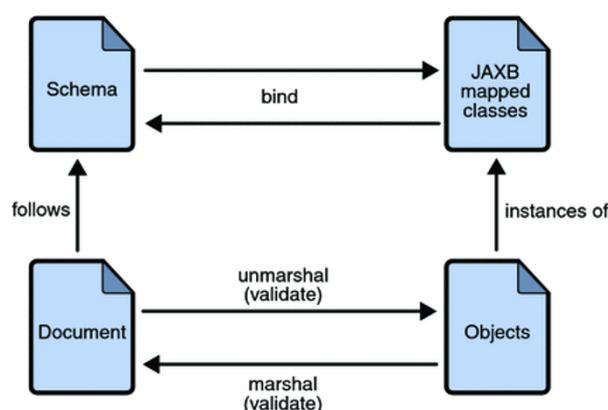
¹ O VLC é um reprodutor multimídia livre, de código aberto, multi-plataforma, e um arcabouço que reproduz a maioria dos arquivos de mídia, bem como DVD, CD de áudio, VCD e vários protocolos de fluxo de rede (VIDEOLAN, 2015).

MPEG Metadata², que contém todos os objetos possíveis do padrão MPEG-V (parte 1 até 5) e parte dos padrões MPEG-7 e MPEG-21, possibilitando o desempacotamento automático do XML na aplicação por meio da API JAXB (*Java Architecture for XML Binding*).

Para definir a estrutura e regras que um XML deve seguir e validá-lo, um arquivo XSD (*XML Schema Definition*) pode ser criado. Por exemplo, é possível definir quais atributos a tag `<effect>` da parte 3 do MPEG-V pode conter. A partir da ferramenta Eclipse, é possível gerar classes Java a partir de esquemas XSD e vice-versa, usando também a API JAXB. Então, os esquemas XSD disponibilizados pelo grupo MPEG foram transformados em objetos Java dando origem a essa biblioteca.

A Figura 23 apresenta o processo de ligação utilizando JAXB:

Figura 23 – Processo de ligação JAXB.



Fonte: (ORACLE, 2015b)

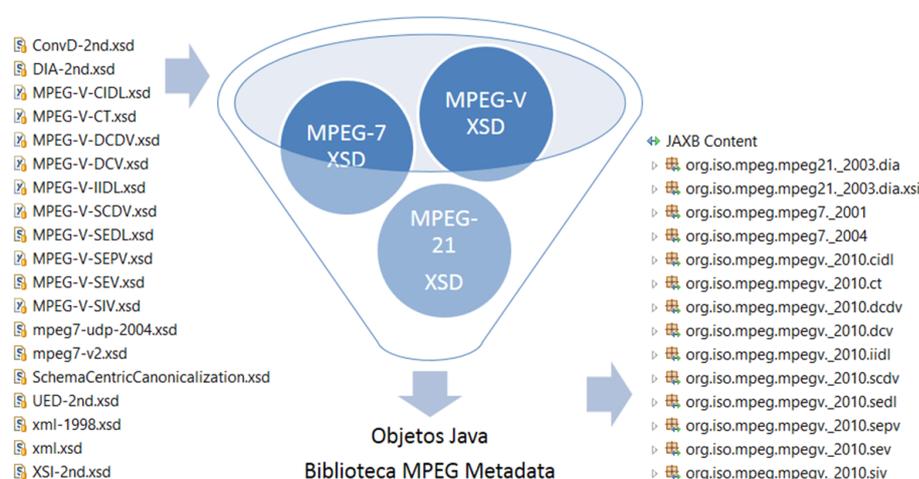
Resumidamente, as etapas do processo de ligação com o JAXB são:

- Gerar classes: um esquema XSD é usado como entrada para gerar as classes;
- *Unmarshal*: documentos XML escritos de acordo com as restrições no esquema de origem são desempacotados pelo JAXB, dando origem a uma árvore de conteúdo;
- Validar (opcional): Se a árvore for modificada, o JAXB pode validar o conteúdo do XML de acordo com a nova árvore;
- Processar dados: os dados do XML podem ser lidos ou modificados com as classes geradas para ligação com o JAXB;
- *Marshal*: documentos XML são criados por meio do empacotamento a partir das classes mapeadas com o JAXB.

² MPEG Metadata: disponível em <<https://github.com/estevaosaleme/MpegMetadata>>

O método é semelhante ao processo utilizado por [Waltl et al. \(2013\)](#) na ferramenta de autoria SEVino e no simulador SESim. Os autores também transformaram esquemas XSD em objetos, porém ao invés de criar os objetos diretamente na aplicação, para o PlaySEM foi criada uma biblioteca que pode ser reutilizada, evitando assim a manutenção de 2 códigos paralelos. Outra diferença é o suporte a todas as partes do MPEG-V, restringida no trabalho de [Waltl et al. \(2013\)](#) somente ao vocabulário SEV e a linguagem SEDL. Essa restrição impediria, por exemplo, a troca de informações sobre a capacidade dos dispositivos, que utiliza a linguagem CIDL e o vocabulário DCDV, apresentados no capítulo 2. A Figura 24 ilustra o processo de criação da biblioteca MPEG *Metadata*.

Figura 24 – Processo de criação da biblioteca MPEG *Metadata*.



Fonte: Elaborada pelo autor

4.2 A Plataforma PlaySEM compatível com MPEG-V

4.2.1 Arquitetura do sistema

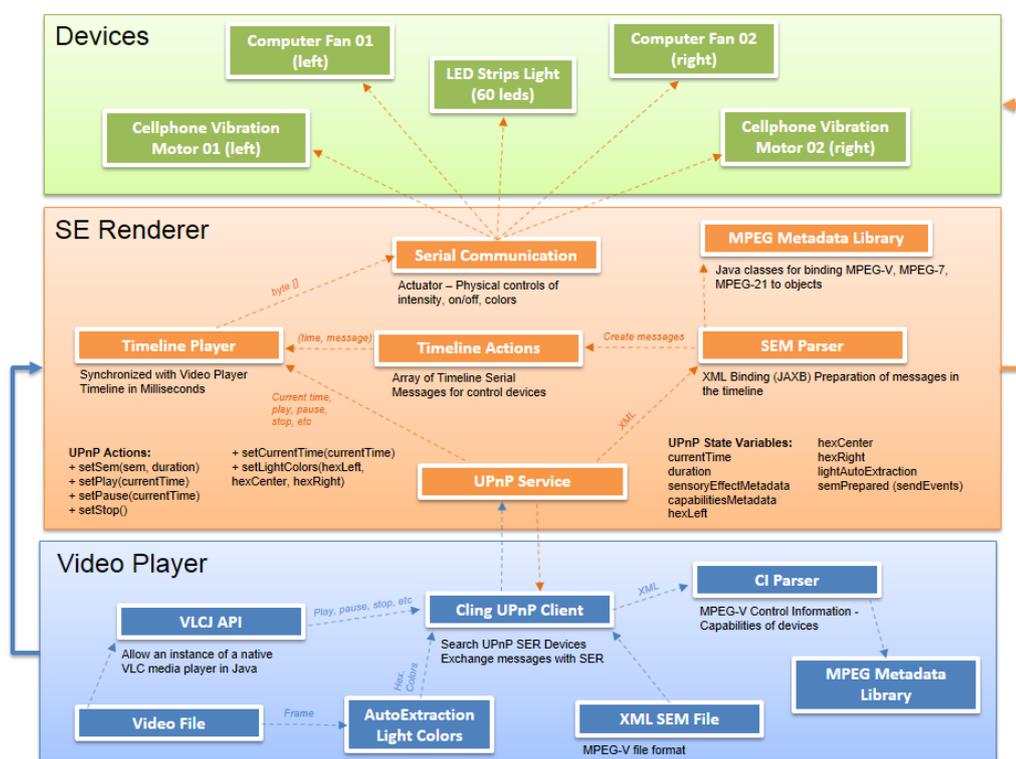
As representações de arquiteturas de software fornecem uma visão holística a partir de abstrações que representam a estrutura e organização dos componentes, bem como seus relacionamentos ([PRESSMAN; MAXIM, 2014](#)). A ISO/IEC 42010:2011 ([ISO/IEC/IEEE. . . , 2011](#)), um padrão internacional para descrições de arquitetura de sistemas e software, define arquitetura como “*conceitos ou propriedades fundamentais de um sistema em seu ambiente, personificados em seus elementos, relacionamentos, e nos princípios de seu projeto e evolução*”.

De acordo com [Sommerville \(2011\)](#), o projeto arquitetural é o primeiro estágio do processo de *design* do software. Ele ajuda a identificar os principais componentes estruturais. Como saída, a fase de *design* deve descrever como o projeto está organizado através de um conjunto de componentes que se comunicam. [Sommerville \(2011\)](#) afirma

ainda que os diagramas de blocos apresentam uma imagem de alto nível do sistema que permite que pessoas de diferentes disciplinas possam compreender o sistema com maior facilidade.

Parece ser um consenso em engenharia de software que o projeto arquitetural de um sistema provê agilidade na disseminação de conhecimento sobre a sua estrutura. Nessa direção, a Figura 25 apresenta a estrutura e relacionamentos entre os componentes da plataforma PlaySEM. A arquitetura está dividida em 3 camadas: SE *Video Player*, SE *Renderer* e *Devices*. Essa divisão fornece vantagens possibilitando que uma camada possa ser substituída por outra equivalente. Além disso, quando é necessário alterar ou adicionar um serviço em alguma camada, apenas a camada adjacente é afetada [Sommerville \(2011\)](#). De modo análogo ao conceito de divisão de camadas apresentado por [Pressman e Maxim \(2014\)](#), na plataforma PlaySEM cada camada realiza operações que progressivamente se tornam mais próximo do conjunto de instruções de máquina (SE *Video Player* -> SE *Renderer* -> *Devices*).

Figura 25 – Estrutura e relacionamentos dos componentes da plataforma PlaySEM.



Fonte: Elaborada pelo autor

Cada camada retratada na Figura 25 é responsável por funções bem definidas na plataforma PlaySEM. O Quadro 3 descreve as funcionalidades de cada um dos componentes da arquitetura.

A camada SE *Video Player* representa, como o próprio nome indica, funções

Quadro 3 – Função dos componentes da arquitetura da plataforma PlaySEM.

Camada	Componente	Função
SE <i>Video Player</i>	<i>Video File</i>	Arquivo de entrada de conteúdo audiovisual. É manuseado pela API VLCJ.
	VLCJ API	Permite instanciar o reprodutor de mídias VLC no ambiente da linguagem Java. Quando ocorre eventos como <i>Play</i> , <i>Pause</i> ou <i>Stop</i> , pode se comunicar com o componente <i>Cling UPnP Client</i> para sincronização de conteúdo com os dispositivos físicos.
	XML SEM <i>File</i>	Arquivo de metadados de efeitos sensoriais. É encaminhado pelo componente <i>Cling UPnP Client</i> para a camada SE <i>Renderer</i> .
	<i>AutoExtraction Light Colors</i>	Capaz de capturar o <i>frame</i> do vídeo e calcular as cores médias de cada lado do quadro, enviando em tempo real para a camada SE <i>Renderer</i> , por meio do componente <i>Cling UPnP Client</i> , as cores para serem renderizadas no dispositivo de luz.
	<i>CI Parser</i>	Responsável por transformar as informações sobre as capacidades dos dispositivos do SE <i>Renderer</i> , descritas no formato MPEG-V Parte 2, em objetos Java. É auxiliado pelo componente <i>MPEG Metadata Library</i> .
	<i>MPEG Metadata Library</i>	Facilita a ligação entre os metadados no formato MPEG-V e objetos da aplicação.
	<i>Cling UPnP Client</i>	Efetuar a procura por dispositivos SE <i>Renderer</i> 's na rede local e prover a comunicação UPnP entre o SE <i>Video Player</i> e os serviços do SE <i>Renderer</i> .
SE <i>Renderer</i>	<i>UPnP Service</i>	Fornece um barramento de serviços através da interface UPnP incluindo o processamento do SEM e controle sincronizado dos dispositivos.
	<i>SEM Parser</i>	Recebe um arquivo SEM como entrada e transforma em um ou mais comandos para que os dispositivos físicos renderizem os efeitos sensoriais, alinhando as ações em uma linha de tempo da duração da mídia.
	<i>Timeline Actions</i>	Armazenar as mensagens serializadas em <i>bytes</i> para acionamento dos dispositivos em uma linha de tempo.
	<i>Timeline Player</i>	Linha de tempo com duração da mídia, sincronizada com o SE <i>Video Player</i> .
	<i>Serial Communication</i>	Aciona os atuadores com parâmetros de intensidade, cor, liga/desliga, etc, para renderização dos efeitos sensoriais.
	<i>MPEG Metadata Library</i>	Facilita a ligação entre os metadados no formato MPEG-V e objetos da aplicação.
Devices	<i>Cellphone Vibration Motor 01 (left)</i>	Dispositivo físico para prover efeito de vibração na localização esquerda do usuário.
	<i>Computer Fan 01 (left)</i>	Dispositivo físico para prover efeito de vento na localização esquerda do usuário.
	<i>LED Strips Light (60 leds)</i>	Dispositivo físico para prover efeito de luz na localização esquerda, centro ou direita do usuário.
	<i>Computer Fan 02 (right)</i>	Dispositivo físico para prover efeito de vento na localização direita do usuário.
	<i>Cellphone Vibration Motor 02 (right)</i>	Dispositivo físico para prover efeito de vibração na localização direita do usuário.

Fonte: Elaborado pelo autor

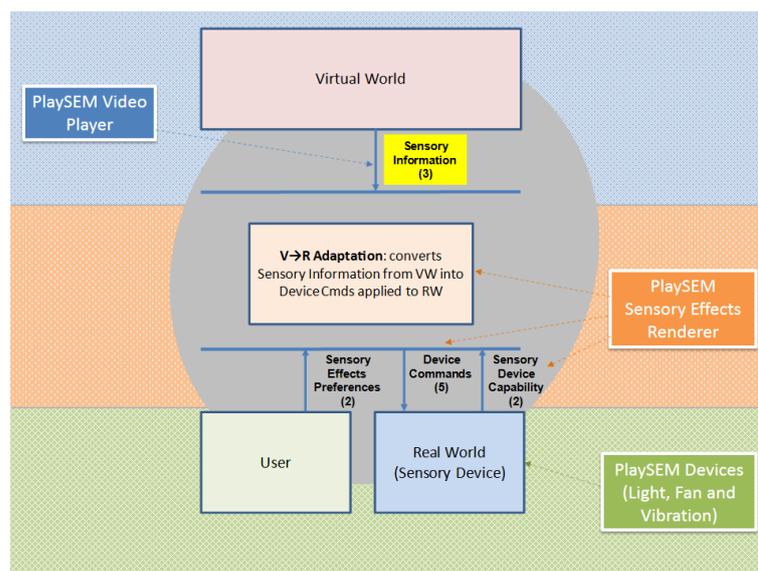
relativas a um *Media Player* para reprodução audiovisual de conteúdo, permitindo como entrada um arquivo SEM, ausente em *media players* tradicionais. Adicionalmente, a camada possui componentes que permitem a comunicação com o SE *Renderer* por meio da interface UPnP.

A camada *SE Renderer* tem como responsabilidade transformar o SEM descrito no formato MPEG-V em um ou mais comandos para renderizar efeitos sensoriais com luz, vento ou vibração, alinhados sob uma mesma linha temporal, além de sincronizar o conteúdo entre o *SE Video Player* com as ações realizadas no dispositivo.

Na camada *Devices* estão situados os dispositivos físicos, que são acionados para fornecer ao usuário os efeitos multissensoriais de luz, vento ou vibração durante a reprodução de um conteúdo anotado em conformidade com o MPEG-V.

Confrontando a arquitetura da parte 3 do MPEG-V proposta por [Choi e Kim \(2012\)](#) com as camadas da plataforma PlaySEM, pode-se considerar a camada *SE Video Player* como representante do mundo virtual (*Virtual World*), a camada *SE Renderer* como representante do motor de adaptação do mundo virtual para real (*V->R Adaptation*) e a camada *Devices* como representante do mundo real (*Real World*), onde estão situados os dispositivos físicos. A Figura 26 exhibe esse cenário.

Figura 26 – Camadas da plataforma PlaySEM adaptadas na arquitetura da parte 3 do MPEG-V proposta por Choi e Kim.

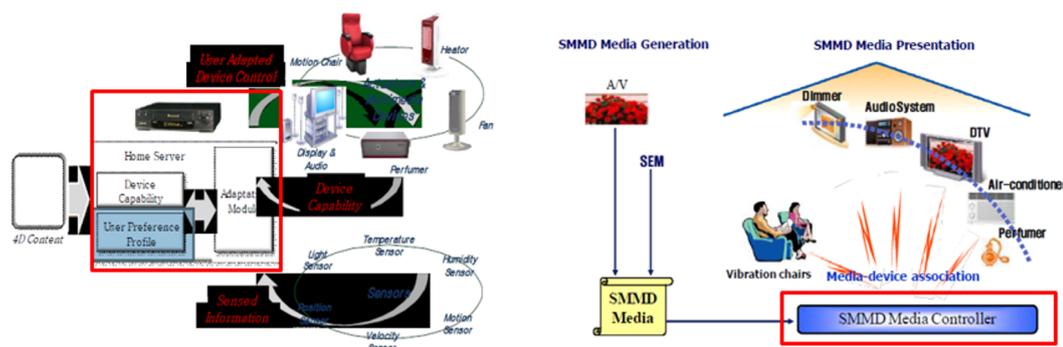


Fonte: Adaptada de ([CHOI; KIM, 2012](#))

A arquitetura proposta também foi influenciada por outros modelos propostos para renderização de efeitos sensoriais. A Figura 27 apresenta esses modelos, destacando em vermelho os renderizadores de efeitos sensoriais, chamado de “Home Server” por [Choi, Lee e Yoon \(2011\)](#) e de “SMMD Media Controller” por [Suk, Hyun e Yong \(2009\)](#).

A arquitetura conceitual proposta por [Choi, Lee e Yoon \(2011\)](#) concentra um servidor em um ambiente doméstico para reproduzir não somente conteúdo audiovisual, mas também orquestrar os atuadores que podem renderizar os efeitos sensoriais. O “Home Server” é capaz de receber e processar metadados de efeitos sensoriais no formato MPEG-V

Figura 27 – Home Server à esquerda e SMMD Media Controller à direita



Fonte: Adaptada de (CHOI; LEE; YOON, 2011) e (SUK; HYUN; YONG, 2009)

e sincronizar o conteúdo audiovisual com as ações nos dispositivos físicos.

O “SMMD Media Controller” apresenta o conceito de procura de dispositivos capazes de apresentar os efeitos de acordo com o SEM, descrito no formato MPEG-V. No processo de apresentação, a mídia SMMD é transmitida para um terminal por serviços de *streaming* ou *upload*.

A arquitetura da plataforma PlaySEM reúne atributos semelhantes dessas duas propostas, como por exemplo, a característica de processamento independente do SEM pelo *SE Renderer* e a procura por dispositivos capazes de prover os efeitos sensoriais. Aliada às técnicas de desenvolvimento de software, a proposta transpõe o modelo conceitual para a prática permitindo validar seus requisitos e avaliar seu desempenho.

4.2.2 PlaySEM SE Video Player

O software *SE Video Player*³ é responsável pela reprodução de vídeo e pela leitura dos metadados de efeitos sensoriais em MPEG-V integrados ao conteúdo do vídeo. Com o intuito de explorar as funcionalidades, a interface e o comportamento do software, essa seção apresenta um diagrama de casos de uso do software, algumas imagens da sua interface gráfica e um diagrama de atividades da principal tarefa do software.

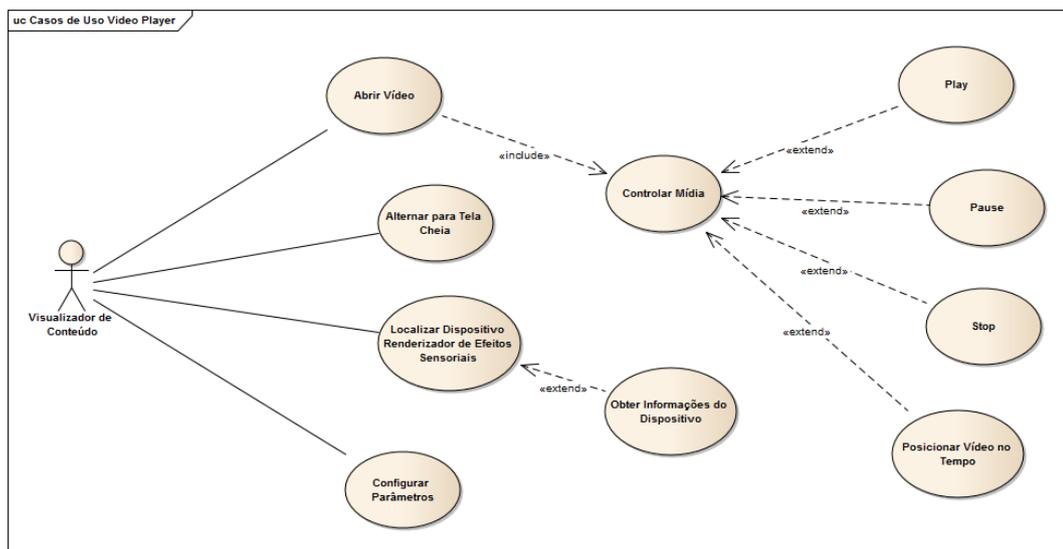
De uma maneira simplificada, a partir de um caso de uso é possível identificar os atores envolvidos e quais funcionalidades são acessíveis a ele em um alto nível de abstração. Sommerville (2011) destaca que diagramas de casos de uso mostram as interações entre o sistema e o ambiente. Em consonância com as afirmações anteriores, Pressman e Maxim (2014) dizem que diagramas de caso de uso UML ajudam a determinar as funcionalidades e recursos do software a partir da perspectiva do usuário, fornecendo uma visão geral de todos os casos de uso e como eles estão relacionados. O ator é um agente externo ao

³ PlaySEM SE Video Player: disponível em <https://github.com/estevaosaleme/PlaySEM_SEVideoPlayer>

sistema, não sendo necessariamente um usuário humano.

A Figura 28 exibe o diagrama de casos de uso na linguagem UML do software SE *Video Player*. O ator é identificado como “Visualizador de Conteúdo”, um usuário fora da fronteira da aplicação que interage com o sistema.

Figura 28 – Diagrama de casos de uso do software SE *Video Player*.



Fonte: Elaborada pelo autor

Cada caso de uso é descrito a seguir para entendimento da funcionalidade do sistema (requisitos funcionais), e não tem intenção de mostrar a interação passo a passo como uma descrição de caso de uso:

- **Abrir Vídeo:** permite ao ator selecionar um arquivo de vídeo no sistema de arquivos. Verifica se o vídeo possui arquivo SEM associado e disponibiliza para o SE *Renderer* caso positivo;
- **Controlar Mídia:** após o carregamento do vídeo, esse caso de uso de inclusão fornece ao ator opções de extensão para controle da mídia;
- **Play:** esse caso de uso de extensão permite ao ator reproduzir o conteúdo audiovisual e acionar o início dos efeitos sensoriais, caso o SE *Renderer* esteja selecionado;
- **Pause:** esse caso de uso de extensão permite ao ator congelar o conteúdo audiovisual e os efeitos sensoriais, caso o SE *Renderer* esteja selecionado;
- **Stop:** permite ao ator parar o conteúdo audiovisual e os efeitos sensoriais, caso o SE *Renderer* esteja selecionado;

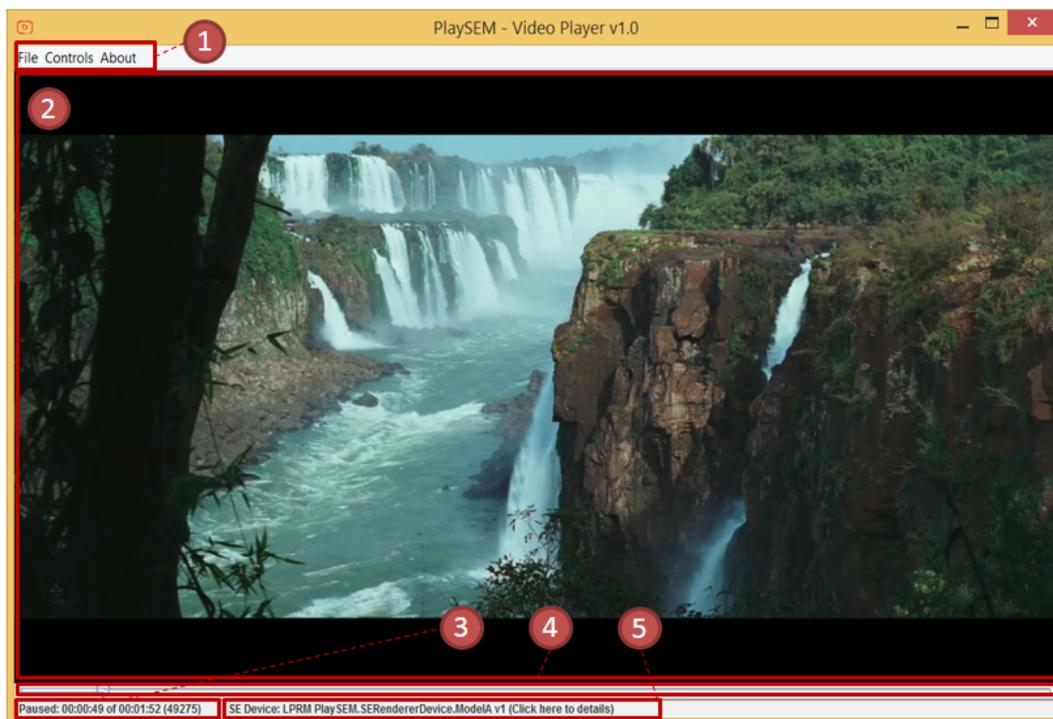
- Posicionar Vídeo no Tempo: esse caso de uso de extensão permite ao ator selecionar a posição desejada do conteúdo audiovisual, sincronizando com as ações controladas pelo SE *Renderer* caso esteja selecionado;
- Alternar para Tela Cheia: como o próprio nome indica, permite ao ator alterar o conteúdo audiovisual para tela cheia;
- Localizar Dispositivo Renderizador de Efeitos Sensoriais: permite ao ator localizar um dispositivo SE *Renderer* na rede local para o processamento do SEM e renderização dos efeitos sensoriais;
- Obter Informações do Dispositivo: esse caso de uso de extensão permite ao ator opcionalmente obter informações sobre a capacidade dos atuadores controlados pelo SE *Renderer* (disponibilizados no formato MPEG-V), indicando, por exemplo, as cores que o atuador de luz é capaz de reproduzir;
- Configurar Parâmetros: permite ao ator configurar parâmetros para indicar o diretório que o VLC está instalado e o tema da aplicação.

A Figura 29 exibe a interface da tela principal do software SE *Video Player*. O item 1, é o menu de opções do usuário. A partir dele é possível abrir arquivo de vídeo, localizar dispositivo renderizador de efeitos sensoriais, configurar parâmetros, controlar a mídia, alterar para modo de tela cheia e obter informações sobre o software. No item 2 o conteúdo audiovisual é reproduzido. O item 3 indica o estado do vídeo (reproduzindo, pausado ou parado) e o tempo decorrido. O item 4 é uma barra deslizante que permite ao usuário avançar ou retroceder o vídeo. O item 5 mostra o dispositivo que foi selecionando permitindo ao usuário obter informações sobre a capacidade dos dispositivos a partir de um clique.

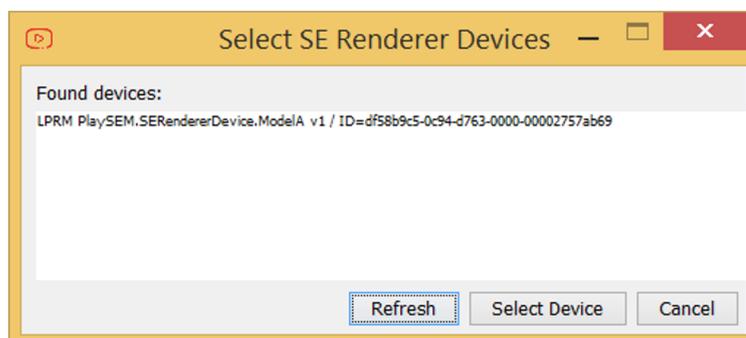
A Figura 30 exibe a interface para selecionar dispositivos do tipo SE *Renderer* disponíveis na rede local. A cada 5 segundos a lista de dispositivos é automaticamente atualizada. Também há uma opção de atualização manual por meio do botão “Refresh”.

A tela exibida na Figura 31 é acionada após o usuário solicitar informações sobre as capacidades dos dispositivos de exibição. O item 1 destaca as informações sobre o dispositivo SE *Renderer* fornecidas pelas propriedades UPnP, como por exemplo, o *id*, o fabricante, o modelo do dispositivo. No item 2 são apresentadas informações sobre as capacidades dos dispositivos obtidos a partir de mensagem UPnP carregando informações no formato MPEG-V parte 2. O item 3 destaca o tempo em milissegundos para verificar a sincronização com o dispositivo, por meio da comparação com o tempo exibido na propriedade “Current Time” do item 1, informações do dispositivo.

Como relatado anteriormente, a principal funcionalidade do SE *Video Player* é reproduzir conteúdo audiovisual sincronizando-o com as ações do SE *Renderer*. Para

Figura 29 – Interface do software SE *Video Player*.

Fonte: Elaborada pelo autor

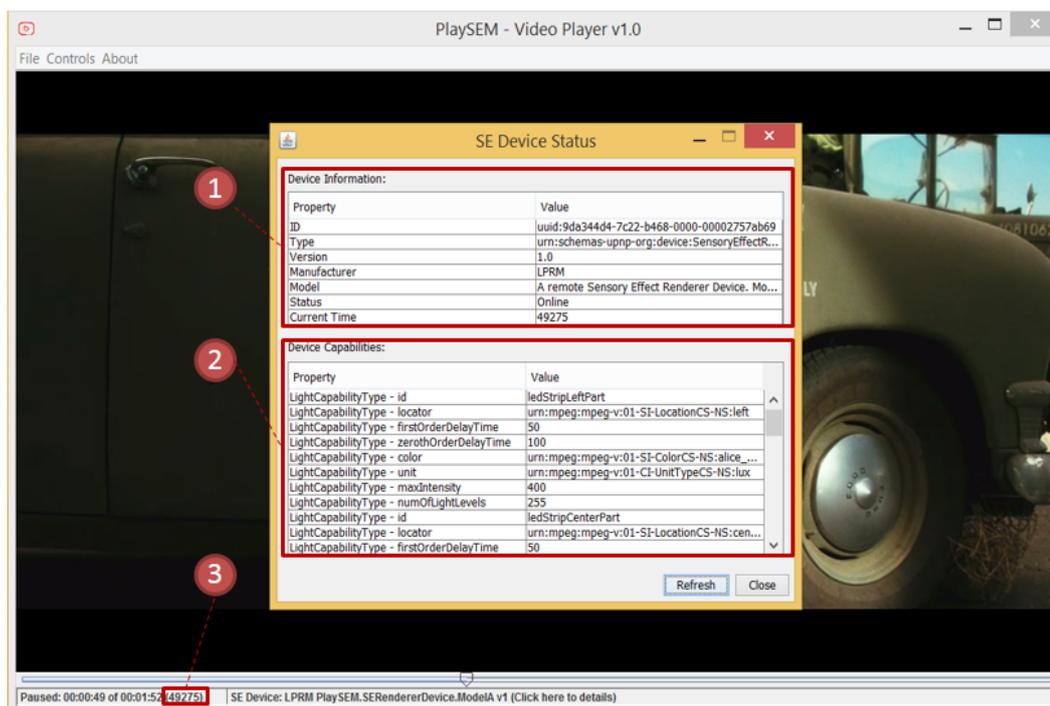
Figura 30 – Procura por dispositivo SE *Renderer* a partir do SE *Video Player* por meio de interface UPnP.

Fonte: Elaborada pelo autor

compreender melhor o comportamento do sistema, o digrama de atividade “Execução de Vídeo” é apresentado na Figura 32. Um diagrama de atividades UML mostra as atividades envolvidas em um processo ou processamento de dados (SOMMERVILLE, 2011). Ele descreve o comportamento dinâmico de um sistema ou parte dele por meio do fluxo de controle entre as ações que o sistema executa (PRESSMAN; MAXIM, 2014).

A atividade “Execução de Vídeo” tem início com um quadro de decisão para que seja possível selecionar um dispositivo renderizador de efeitos sensoriais, caso ainda não tenha sido selecionado. Em seguida, a ação abrir vídeo é executada. Se o vídeo possuir um

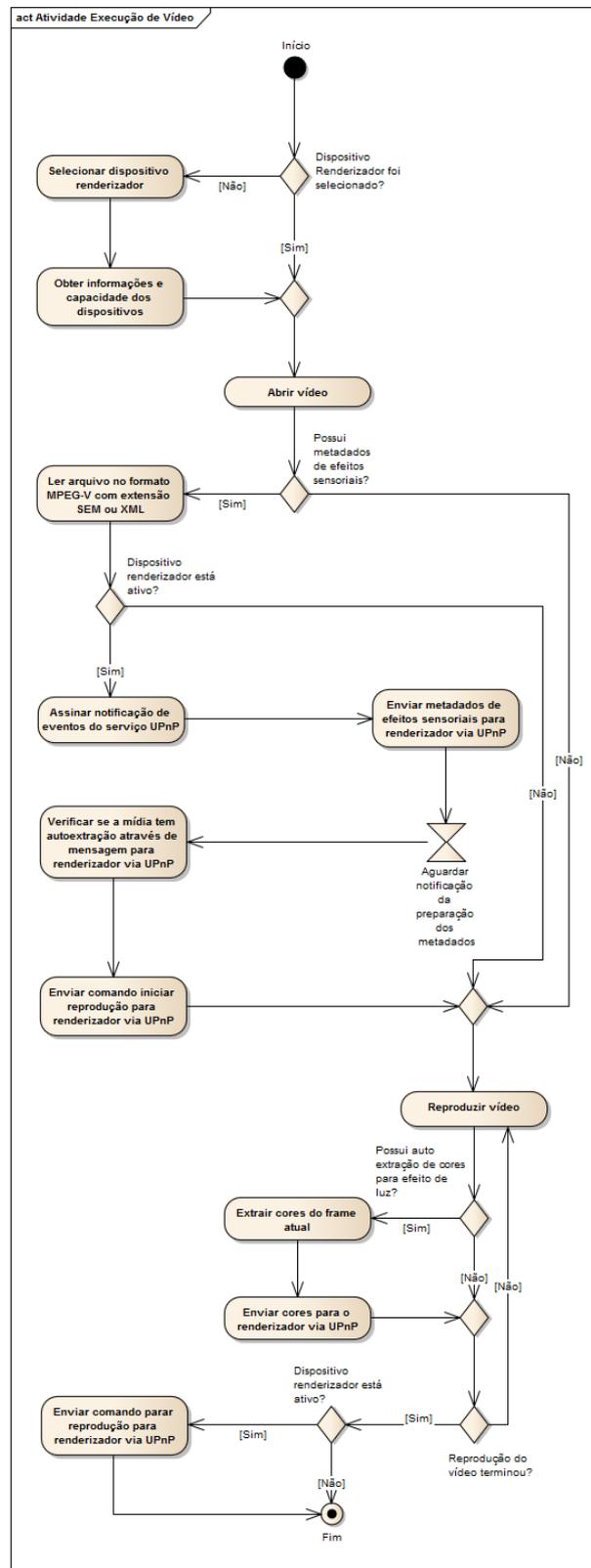
Figura 31 – Descrição das capacidades dos dispositivos.



Fonte: Elaborada pelo autor

arquivo SEM com extensão “.sem” ou “.xml”, então o SE *Video Player* carrega o arquivo para a memória. Se o dispositivo renderizador está ativo, o SE *Video Player* faz assinatura de notificação de eventos UPnP que indicarão a mudança de estado de variáveis marcada como “sendevents=true”. Esse recurso é usado pelo SE *Renderer* para avisar ao SE *Video Player* que o processamento do SEM terminou e está apto para começar a reprodução do vídeo com os efeitos sensoriais. O SE *Video Player* então verifica junto ao SE *Renderer* se o SEM processado possui auto-extração de luz para se preparar para o envio das informações de cores dos *frames* do vídeo. Ao iniciar a reprodução do vídeo, essa sentença é verificada e em caso positivo o SE *Video Player* extrai as cores em três partes (esquerda, centro e direita) e entrega no formato hexadecimal para o SE *Renderer*. Esse processo perdura até que todo o conteúdo do vídeo tenha sido reproduzido. Ao término do processo, caso o dispositivo ainda esteja ativo, um comando para parar o SE *Renderer* e seus dispositivos será executado. Caso o vídeo não tenha arquivo SEM ou o dispositivo SE *Renderer* não esteja ativo, o SE *Video Player* se comporta como um reprodutor de vídeo tradicional, não renderizando os efeitos sensoriais no ambiente do usuário.

Figura 32 – Diagrama de atividades – Execução de Vídeo.



Fonte: Elaborada pelo autor

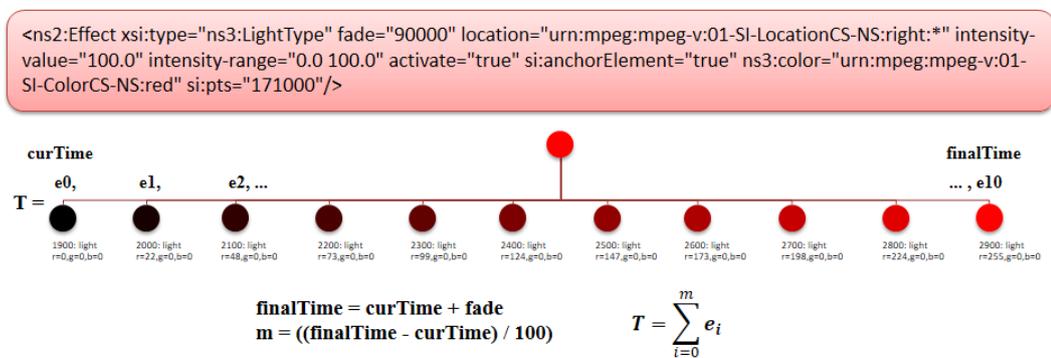
4.2.3 PlaySEM SE *Renderer*

4.2.3.1 Software

O software SE *Renderer*⁴ é responsável pela transformação de metadados de efeitos sensoriais em comandos e acionamento dos dispositivos. É o componente central da arquitetura PlaySEM e pode ser reutilizado com outros *Media Players*, sendo necessária uma adaptação para a comunicação entre os softwares (ver seção 4.3). Vinculado a ele, há um módulo para controle de hardware, gerenciado por um outro software que roda no Arduino. De modo sintético, o SE *Renderer* recebe o SEM e transforma cada elemento no formato MPEG-V em um ou mais comandos para acionar os dispositivos físicos em uma linha de tempo com duração igual à duração do vídeo reproduzido no *Media Player*.

A Figura 33 exemplifica a transformação de uma anotação de um efeito de luz para ações a serem executadas pelos dispositivos físicos. A cor definida para o efeito é vermelha, no lado direito do usuário, com intensidade de 100% e “fade in” equivalente a 1 segundo (considerando $timescale = 90000$) a partir do milissegundo 1900 de execução (171000/90000). Ao fim, 11 ações de cores são geradas pelo SE *Renderer* para criar o efeito de “fade in” até chegar na cor vermelha.

Figura 33 – Transformação de anotação de efeito de luz para comandos com “fade in” de 1 segundo.

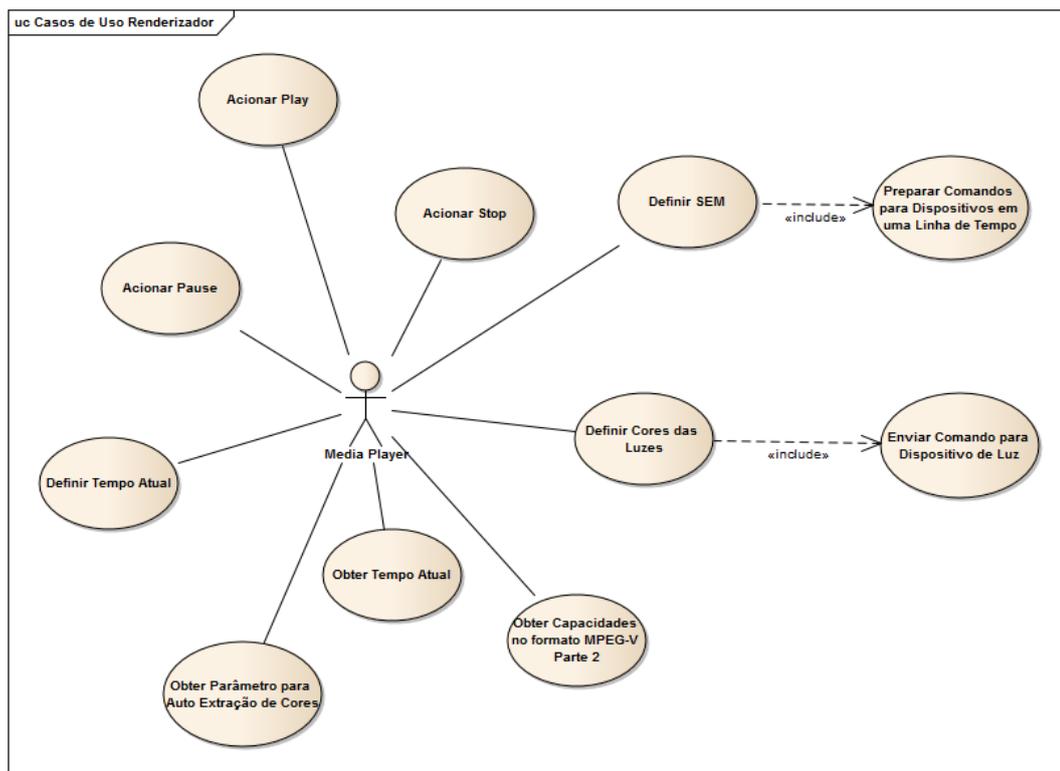


Fonte: Elaborada pelo autor

O agente externo à fronteira do software SE *Renderer*, capaz de se comunicar com seus serviços, é chamado de “Media Player”. A Figura 34 exhibe o diagrama de casos de uso UML com a finalidade de visualizar as funcionalidades providas pelo software.

Assim como o digrama de casos de uso do software SE *Video Player*, não há intenção aqui de se mostrar a interação passo a passo como uma descrição de caso de uso. As funcionalidades suportadas pelo sistema são descritas a seguir.

⁴ PlaySEM SE *Renderer*: disponível em <https://github.com/estevasaleme/PlaySEM_SERenderer>

Figura 34 – Diagrama de casos de uso do software SE *Renderer*.

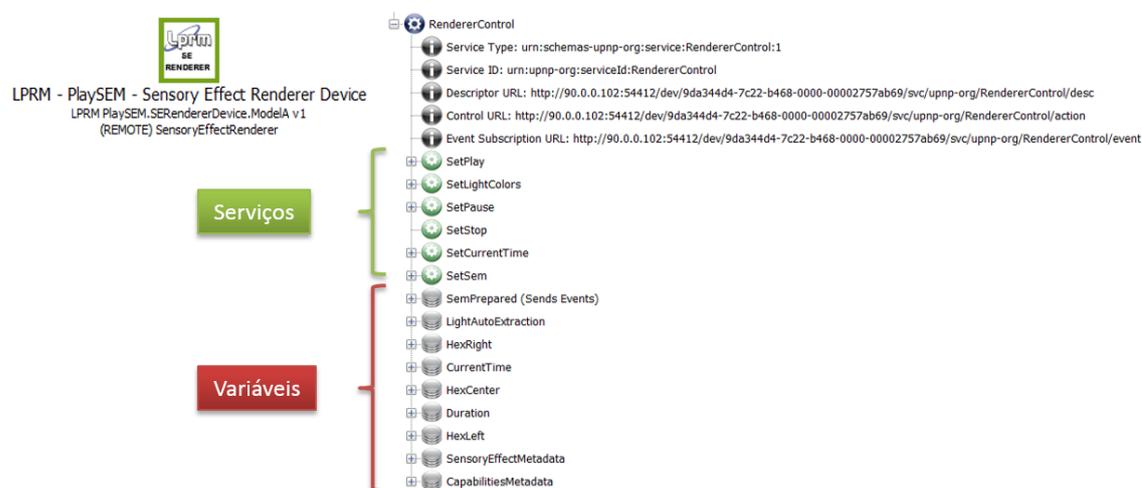
Fonte: Elaborada pelo autor

- Definir SEM: permite ao ator enviar o arquivo de metadados de efeitos sensoriais no formato MPEG-V parte 3 para processamento dos efeitos de luz, vento e vibração;
- Preparar Comandos para Dispositivos em uma Linha de Tempo: após definir o SEM, esse caso de uso de inclusão é executado para dispor os comandos para os dispositivos físicos em uma linha de tempo;
- Acionar *Play*: permite ao ator indicar o início da reprodução do vídeo, para que a renderização dos efeitos sensoriais também seja iniciada;
- Acionar *Pause*: permite ao ator congelar a renderização dos efeitos sensoriais a qualquer instante;
- Acionar *Stop*: permite ao ator parar a renderização dos efeitos sensoriais a qualquer instante;
- Definir Tempo Atual: permite ao ator indicar a posição atual da reprodução do vídeo, para que a renderização dos efeitos sensoriais seja sincronizada;
- Obter Tempo Atual: permite ao ator obter o tempo decorrido na linha de tempo do renderizador de efeitos sensoriais;

- Obter Parâmetro para Auto Extração de Cores: permite ao ator checar se o SEM processado indica que auto-extração de cores deve ser automática;
- Obter Capacidades no formato MPEG-V Parte 2: permite ao ator obter informações sobre a capacidade dos dispositivos atuadores acoplados ao SE *Renderer* no formato MPEG-V parte 2;
- Definir Cores das Luzes: permite ao ator definir em tempo real as cores extraídas das partes esquerda, central e direita do *frame* do vídeo para que o SE *Renderer* encaminhe para o atuador de efeito de luz;
- Enviar Comando para Dispositivo de Luz: após definir as cores, esse caso de uso de inclusão envia os comandos para o dispositivo físico para renderizar o efeito de luz;

Os casos de uso relatados no SE *Renderer* são acionados por meio de serviços e variáveis expostas através da interface UPnP. A Figura 35 apresenta partes da tela do software Cling Workbench, que é capaz de encontrar e exibir dispositivos compatíveis com UPnP. É possível notar a presença de seis serviços e nove variáveis, cuja às descrições são apresentadas nos quadros 4 e 5.

Figura 35 – Cling Workbench - visualização dos serviços e variáveis do SE *Renderer*.



Fonte: Elaborada pelo autor

O diagrama de atividades UML exibido na Figura 36 ajuda a entender o processo de preparação dos efeitos sensoriais. Ele é disparado a partir do serviço “SetSem”.

O processo de preparação dos efeitos sensoriais recebe o arquivo SEM e a duração do vídeo. Em seguida, o SEM é desempacotado para as classes da aplicação, para facilitar o manuseio e iteração entre os elementos. A interação inicia logo após o desempacotamento, e ocorre enquanto houver elementos do tipo “Effect” ou “GroupOfEffects” não processados. Se o elemento é do tipo grupo de efeitos, então os atributos comuns ao grupo são obtidos,

Quadro 4 – Descrição dos serviços expostos pelo SE *Renderer*.

Serviço	Descrição	Argumentos	Função
SetPlay	Inicia a renderização de efeitos sensoriais, se o SEM estiver setado.	String Current-Time	Indica o tempo para início da execução de efeitos sensoriais.
SetLightColors	Seta as cores do dispositivo de luz em tempo real. Esse serviço responde somente se o estado estiver como “reproduzindo” e o SEM indicar auto-extração de cores do vídeo.	String HexLeft	Cor no formato hexadecimal para ser reproduzida no lado esquerdo do ambiente do usuário.
		String HexCenter	Cor no formato hexadecimal para ser reproduzida no centro do ambiente do usuário.
		String HexRight	Cor no formato hexadecimal para ser reproduzida no lado direito do ambiente do usuário.
SetPause	Congela a renderização de efeitos sensoriais, se o SEM estiver setado.	String Current-Time	Indica o tempo para o congelamento da execução de efeitos sensoriais.
SetStop	Para a renderização de efeitos sensoriais, se o SEM estiver setado.	String Current-Time	Indica o tempo para parar a execução de efeitos sensoriais.
SetCurrentTime	Posiciona o cursor de execução do SE <i>Renderer</i> na linha de tempo, se o SEM estiver setado.	String Current-Time	Indica o tempo para posição do cursor na linha de tempo da execução de efeitos sensoriais.
SetSem	Processa os metadados de efeitos sensoriais, transformando anotações MPEG-V em comandos para os dispositivos.	String SensoryEffectMetadata	Arquivo SEM no formato MPEG-V Parte 3.
		String Duration	Indica a duração do vídeo, que será a mesma duração da linha de tempo da execução de efeitos sensoriais.

Fonte: Elaborado pelo autor

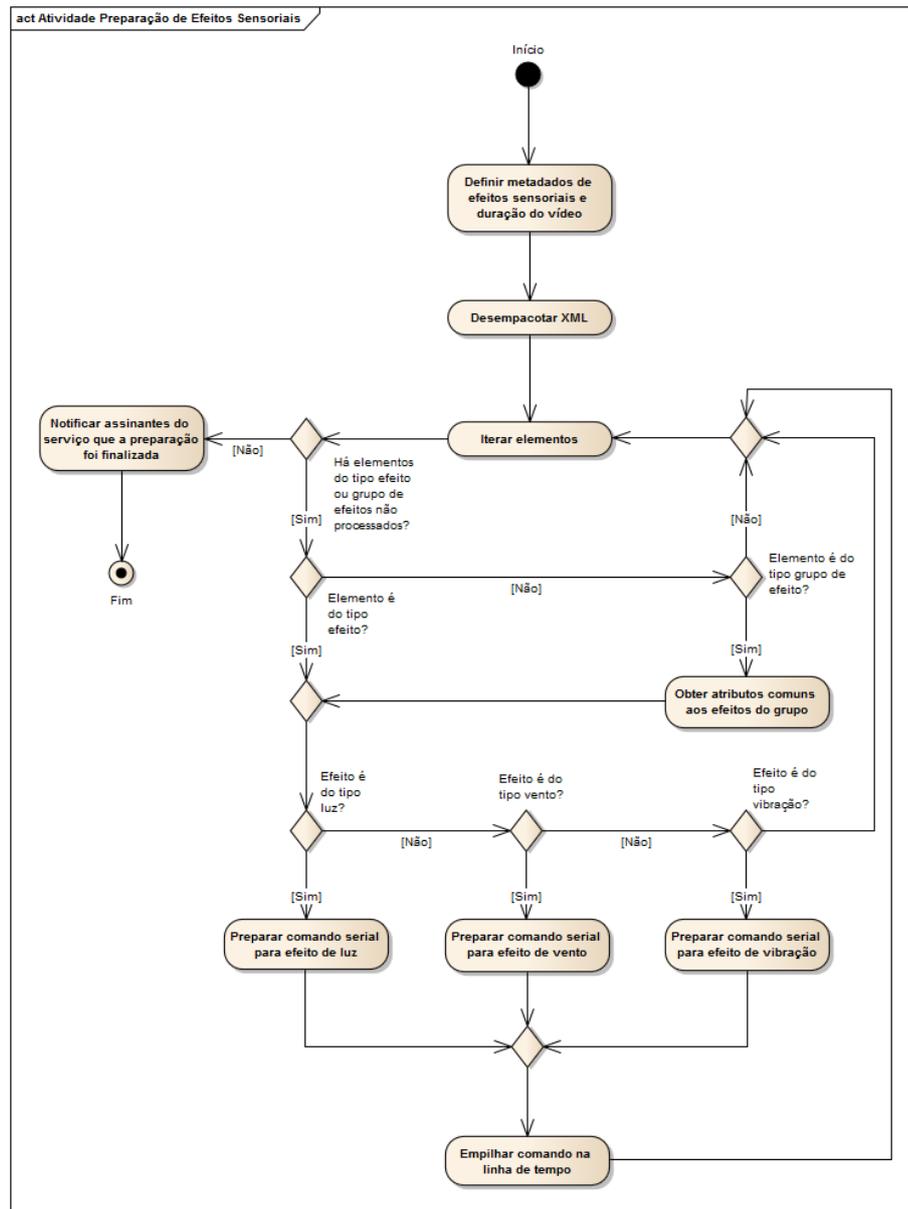
como por exemplo o tempo de início. Se o efeito é do tipo luz, então uma ou mais mensagens no formato de *bytes* para se comunicar com o dispositivo de luz é preparado, considerando os atributos anotados. O mesmo ocorre para os tipos de vento e vibração. Na sequência,

Quadro 5 – Descrição das variáveis expostas pelo SE *Renderer*.

Variável	Tipo de retorno	Descrição
SemPrepared	Boolean	Indica se o SEM foi processado. Possui atributo “sendevents=true”, para que o SE <i>Renderer</i> notifique automaticamente a mudança de estado na variável aos assinantes do serviço.
LightAutoExtraction	Boolean	Indica se o SEM processado está anotado com auto-extração de cores para efeito de luz.
HexRight	String	Retorna a cor setada atual no lado direito do atuador de luz, no formato hexadecimal.
CurrentTime	String	Retorna o tempo atual decorrido no SE <i>Renderer</i> .
HexCenter	String	Retorna a cor setada atual no centro do atuador de luz, no formato hexadecimal.
Duration	String	Retorna a duração total da linha de tempo dos efeitos sensoriais.
HexLeft	String	Retorna a cor setada atual no lado esquerdo do atuador de luz, no formato hexadecimal.
SensoryEffectMetadata	String	Retorna o SEM enviado para o SE <i>Renderer</i> no formato MPEG-V parte 3.
CapabilitiesMetadata	String	Retorna as capacidades dos dispositivos atuadores de efeito sensoriais no formato MPEG-V parte 2.

Fonte: Elaborado pelo autor

Figura 36 – Diagrama de atividades – Preparação de efeitos sensoriais.



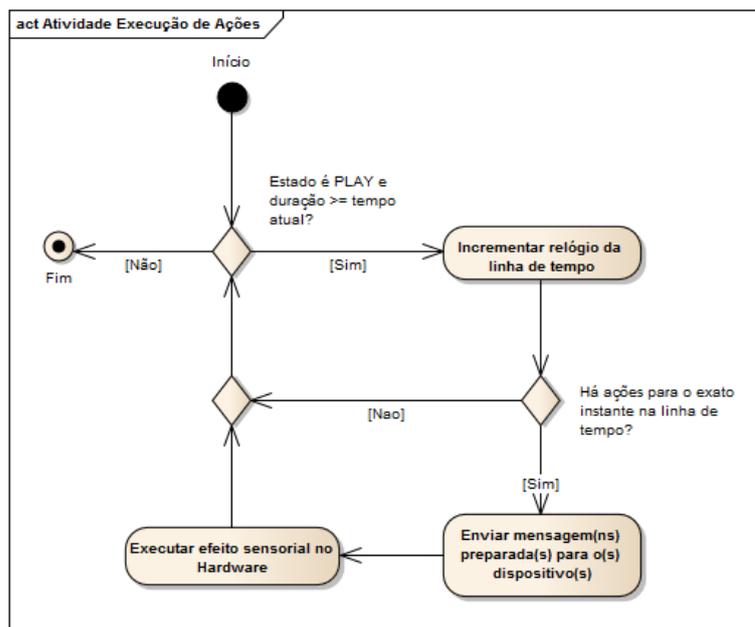
Fonte: Elaborada pelo autor

as mensagens preparadas são empilhadas na linha de tempo do renderizador de efeitos sensoriais e a condição de *loop* é verificada. Quando não existirem mais elementos a serem processados, o SE *Renderer* notifica aos assinantes do serviço que a preparação dos efeitos sensoriais foi finalizada, permitindo assim a reprodução do vídeo em conjunto da execução das ações preparadas na linha de tempo.

Por fim, o diagrama de atividades UML de execução de ações permite compreender o ciclo de execução dos comandos que produzirão os efeitos sensoriais, preparados a partir do serviço “SetSem”.

Na atividade de execução de ações do SE *Renderer*, enquanto o estado do renderi-

Figura 37 – Diagrama de atividades – Execução de ações.



Fonte: Elaborada pelo autor

zador for igual a “PLAY” (reproduzindo) e a duração for maior ou igual ao tempo atual, o relógio é incrementado e é verificada na pilha de mensagens da linha de tempo se há alguma ação para ser executada naquele instante. Em caso positivo, o SE *Renderer* envia a mensagem para o Arduino para que o efeito seja renderizado no ambiente do usuário.

4.2.3.2 Hardware

A parte de hardware do SE *Renderer* é composto, atualmente, por um Arduino programado para receber como entrada mensagens em *bytes* para acionar uma fita de LEDs, motores de vibração obtidos de aparelhos celulares inutilizados e ventiladores (PC *Fan*) utilizados para dissipação de calor de computadores. Esse conjunto de equipamentos é responsável por renderizar, respectivamente, efeitos de luz, vibração e vento. A conexão entre o SE *Renderer* e o Arduino ocorre por meio de conexão serial.

De acordo com [McRoberts \(2011\)](#), o Arduino é “*um pequeno computador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele*”. É um sistema que pode relacionar-se com o ambiente físico por meio de hardware e software. [McRoberts \(2011\)](#) afirma ainda que a facilidade de utilização é a principal vantagem do Arduino em relação às demais plataformas para desenvolvimento de microcontroladores. Isso permite que usuários não ligados diretamente com a área técnica possam assimilar rapidamente o básico para criar projetos em tempo relativamente pequeno.

O quadro 6 exhibe as características dos dispositivos de hardware do SE *Renderer*

com a respectiva função de cada um.

Quadro 6 – Hardwares acoplados ao SE *Renderer* para renderização de efeitos de luz, vento e vibração.

Hardware	Descrição	Função
	Microcontrolador: ATmega328; 5V; Pinos de E/S digital: 14 (dos quais 6 oferecem saída PWM); Memória <i>Flash</i> : 32 KB	Receber comandos do Renderizador por meio de mensagens no formato de <i>bytes</i> e acionar os dispositivos físicos (Fita de LED endereçável, Motores de vibração e Ventiladores)
	Fita de LED endereçável de 1 metro com 60 LEDs RGB 24 <i>bits</i> ; Controlador WS2811; 8 <i>bits</i> de dados vermelhos, verdes e azuis; 5V	Reproduzir efeitos de cores no ambiente do usuário, auto-extraídas ou manualmente.
	Motor de corrente contínua (DC) Voltagem: 1.5 3V	Reproduzir efeitos de vibração no ambiente do usuário.
	Velocidade máxima do ventilador: 4300 RPM; Conexão de alimentação: 3 pinos; 12V	Reproduzir efeitos de vento no ambiente do usuário.

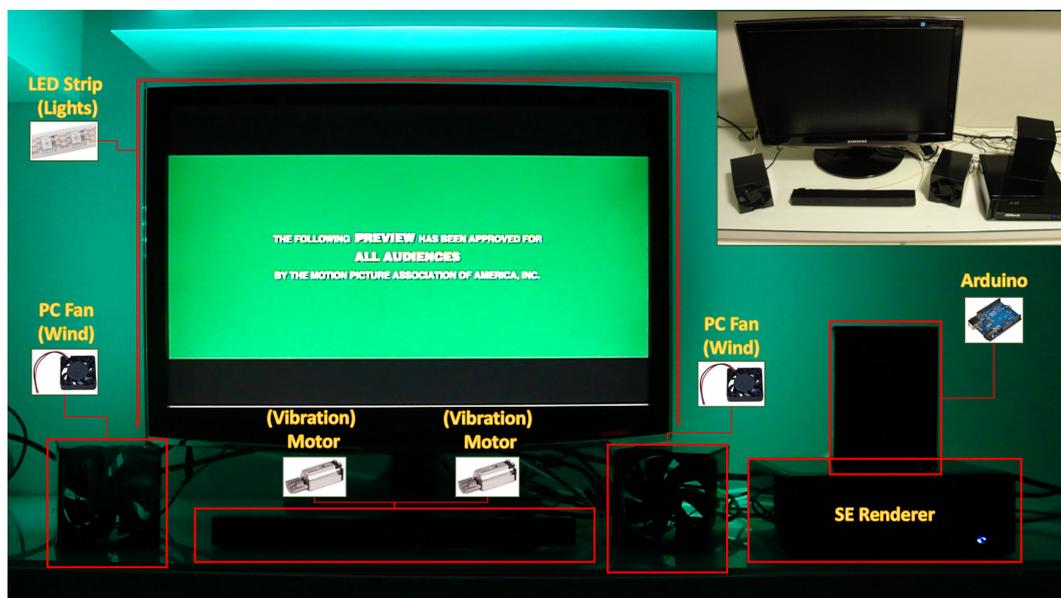
Fonte: Elaborado pelo autor

No *loop* do programa do SE *Renderer* embutido no Arduino, a todo instante é verificado o tipo de comando fornecido como entrada. Então o primeiro caractere é extraído para verificar qual tipo de dispositivo será acionado. Os parâmetros de entrada são descritos a seguir:

- Controlar Fita de LED: formato LICRGB, onde L - indica que o comando é para o dispositivo fita de LED, I - indica o índice do LED que será acionado de 1 a 60, C - indica quantos LEDs a partir do índice I devem ser acionados, R - a cor vermelha de 0 a 255, G - a cor verde de 0 a 255, e B - a cor azul de 0 a 255.
- Controlar Motor de Vibração: formato VIX, onde V - indica que o comando é para o dispositivo motor de vibração, I - o índice do motor (1 esquerda ou 2 direita), e X - indica a intensidade da vibração em uma escala de 0 a 255.
- Controlar Ventilador: formato FIX, onde F - indica que o comando é para o dispositivo ventilador, I - o índice do ventilador (1 esquerda ou 2 direita), e X - indica a intensidade da vibração em uma escala de 0 a 255.

A Figura 38 mostra o conjunto de hardwares relatados no ambiente real do usuário funcionando em conjunto de um mini-pc executando o *SE Renderer* e um monitor apresentando a saída de um computador executando um vídeo no *SE Video Player*.

Figura 38 – Hardwares no ambiente real do usuário.



Fonte: Elaborada pelo autor

Vale ressaltar que as intensidades do motor de vibração e do *PC Fan* são reguladas por meio de tensão e há limitação para reprodução de efeitos com intensidade abaixo de 30% da tensão máxima nos hardwares especificados. Outro inconveniente pode ser o sinal sonoro dos ventiladores em tensões reduzidas.

4.3 Adaptação de *Media Players*

4.3.1 Recomendações gerais

Assim como o *SE Video Player*, qualquer outro *Media Player* pode fazer uso do renderizador de efeitos sensoriais PlaySEM *SE Renderer*. Isso permitiria, por exemplo, que os *Media Players* integrados a *Smart TV's* ou consoles de videogames fossem capazes de reproduzir mídias anotadas com metadados de efeitos sensoriais sem ter que lidar com a complexidade do processamento dos SEM e controle dos atuadores. Para que isso seja possível, é necessário conhecer os serviços disponibilizados pelo *SE Renderer* e a ordenação temporal entre as mensagens que precisam ser trocadas. No caso de aplicações baseadas em eventos, o processo é simples, sendo necessário somente definir o SEM e acionar o serviço “SetPlay” assim que a notificação do evento for recebida. Por exemplo, quando ocorrer o evento explosão em um jogo, a descrição dos efeitos deve ser enviada para *SE Renderer*, seguido do comando “SetPlay” para iniciar a reprodução dos efeitos.

Na próxima seção são apresentados diagramas de sequência UML que exibem a sequência de interações que ocorrem durante um caso de uso, incluindo as mensagens UPnP que devem ser trocadas. Na direção oposta dos diagramas que enfatizam a estrutura estática do programa, como diagrama de classes, o diagrama de sequência é usado para exibir as comunicações dinâmicas entre os objetos durante a execução da tarefa (PRESSMAN; MAXIM, 2014).

Antes de relatar a troca de mensagens, algumas recomendações básicas são listadas para o bom funcionamento da integração:

- O *Media Player* deve estar conectado na mesma rede que o *SE Renderer* e a interface UPnP deve estar habilitada no sistema operacional;
- As *strings* dos argumentos dos serviços UPnP devem ser codificadas com UTF-8;
- A ordenação temporal das mensagens exibidas nos diagramas de sequência deve ser seguida exatamente conforme descrito, para garantir o desempenho mínimo do *SE Renderer*;
- Por questões de segurança, não é recomendável disponibilizar o *SE Renderer* em redes públicas;
- Se o *Media Player* oferecer barra deslizante para posicionar o vídeo no tempo desejado pelo usuário, é recomendável que a mensagem UPnP seja disparada somente após a liberação da seleção na barra, evitando mensagens desnecessárias ao *SE Renderer*;

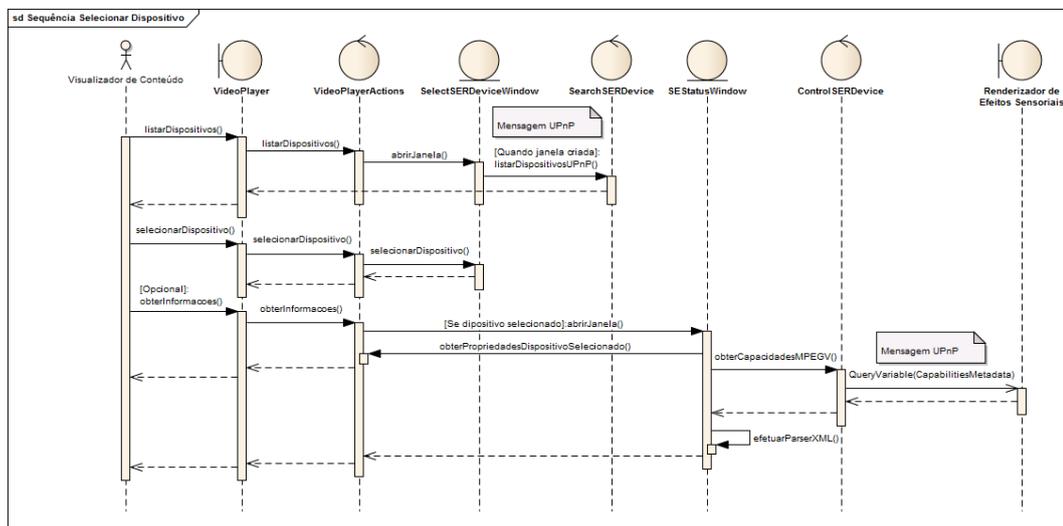
4.3.2 Diagramas para troca de mensagens

Os diagramas de sequência UML apresentados nessa seção seguem o padrão de design ECB (*Entity, Controller, Boundary*). Larman (2004) define “Boundary” como abstrações da interface, “Entity” como objetos independentes da aplicação (tipicamente persistente) e “Controller” como objetos usados para controlar o fluxo das interações do caso de uso.

As entidades (*Entities*) são elementos passivos de longa vida mas não quer dizer necessariamente que são dados e os outros padrões funções. Elas executam algum comportamento organizado em torno de dados e só podem ser acionados por controladores (*Controllers*) ou outras entidades. Os controladores gerenciam o fluxo de interação do cenário, se comunicando com qualquer outro objeto do padrão. Suas preocupações são comportamentos e regras de negócio associados ao cenário. As fronteiras (*Boundaries*) delimitam a entrada ou saída de uma interação com um agente externo e se comunicam internamente somente com os controladores.

A Figura 39 apresenta o diagrama de sequência UML a partir do cenário de uso “Selecionar dispositivo”, que mostra a interação para listar os dispositivos do tipo SE *Renderer* na rede local. O visualizador de conteúdo envia uma mensagem para listar dispositivos para a fronteira *VideoPlayer*, que por sua vez encaminha a mensagem para a classe controladora *VideoPlayerActions*. Então a classe *SelectSERDeviceWindow* é invocada e envia uma mensagem para que a classe *SearchSERDevice* liste os dispositivos UPnP. Em seguida, o ator seleciona o dispositivo. Opcionalmente, ele pode obter informações acerca dos dispositivos, trafegando a solicitação até o renderizador de efeitos sensoriais, onde uma mensagem UPnP assíncrona é enviada. Ao receber os dados, o controlador *SEStatusWindow* efetua o *parser* das capacidades no formato MPEG-V e retorna para a interface do usuário.

Figura 39 – Diagrama de sequência – Selecionar dispositivo.

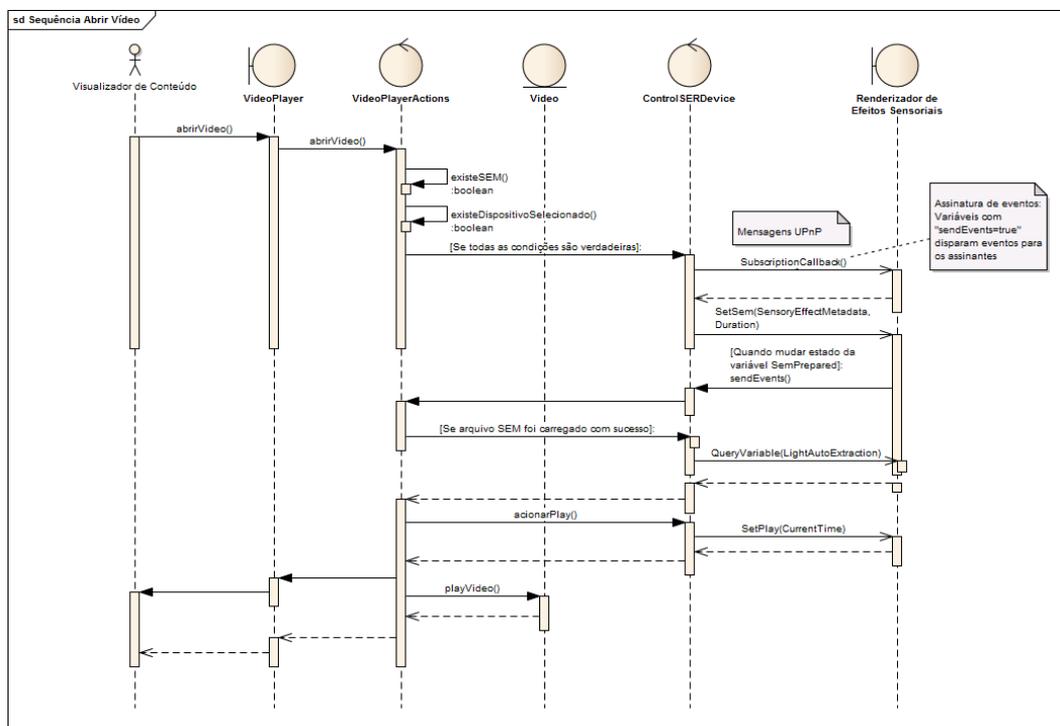


Fonte: Elaborada pelo autor

A sequência “Abrir Vídeo” exibida na Figura 40 ajuda na compreensão dos passos necessários para detectar se o vídeo possui SEM e encaminhá-lo para o SE *Renderer*. O visualizador de conteúdo solicita a abertura de um vídeo qualquer à fronteira *VideoPlayer*. Em seguida, o controlador *VideoPlayerActions* verifica se o vídeo possui SEM associado e se o dispositivo SE *Renderer* foi selecionado. Em caso positivo, o controlador *ControlSERDevice*, responsável por enviar comandos para o SE *Renderer*, envia uma mensagem assíncrona informando que deseja assinar a notificação de eventos do SE *Renderer*. Na sequência, encaminha o SEM por meio da mensagem assíncrona *SetSem* para o renderizador de efeitos sensoriais e fica aguardando a notificação de evento, informando que o SEM foi processado. Após o recebimento do evento, o valor da variável de estado *LightAutoExtraction* é requisitado por meio mensagem UPnP. Ao final do processo o controlador *VideoPlayerActions* solicita ao controlador *ControlSERDevice* que encaminhe uma mensagem UPnP assíncrona para o SE *Renderer*, ordenando a reprodução dos efeitos

sensoriais. A reprodução também é solicitada à entidade *Video* e então o vídeo e os efeitos sensoriais são renderizados no ambiente do usuário.

Figura 40 – Diagrama de sequência – Abrir vídeo.

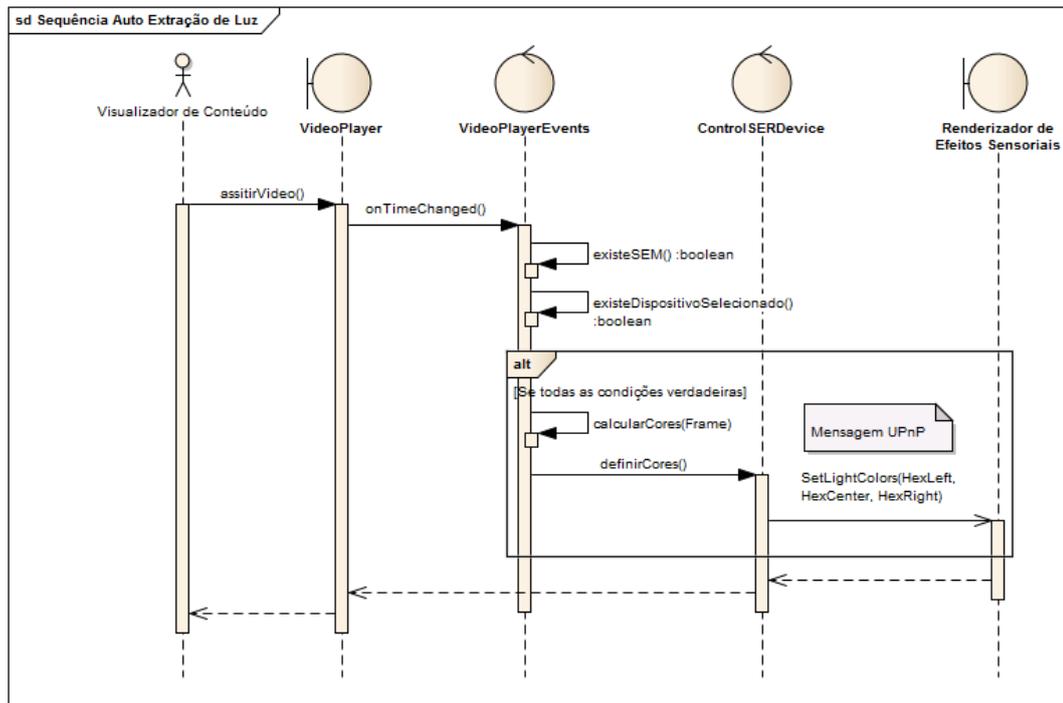


Fonte: Elaborada pelo autor

Quando é necessário que o *Media Player* faça auto extração de cores do vídeo, é requerida o envio de mensagens em fluxo contínuo para o SE *Renderer*. A Figura 41 exibe a interação “Auto extração de luz”. A medida que o evento *onTimeChange* ocorre, caso o vídeo possua SEM associado e o dispositivo SE *Renderer* esteja selecionado, então um método evento para calcular cores a partir do *frame* do vídeo deve ser disparado, extraíndo as cores médias das partes esquerda, direita e central. As cores extraídas são formatadas em hexadecimal e encaminhadas por meio de mensagem UPnP assíncrona para o SE *Renderer*, que por sua vez aciona instantaneamente o atuador de luz. Esse processo deve ocorrer com o menor tempo possível para que o usuário tenha a real sensação das cores do vídeo em seu ambiente.

As sequências “Play”, “Pause”, “Stop” e “Posicionar vídeo no tempo” são muito semelhantes e para ilustrar a interação, somente a sequência *Play* é exibida (Figura 42). É iniciada com uma mensagem *acionarPlay* para a fronteira da aplicação. Caso o vídeo possua SEM associado e o dispositivo SE *Renderer* esteja selecionado, então uma mensagem UPnP assíncrona a partir do controlador *ControlSERDevice* é enviada para o SE *Renderer*. No mesmo instante, o controlador *VideoPlayerActions* solicita à entidade *Video* a reprodução do Vídeo. Por fim, a ação é reproduzida no ambiente do usuário em conjunto dos efeitos sensoriais.

Figura 41 – Diagrama de sequência – Auto extração de luz.

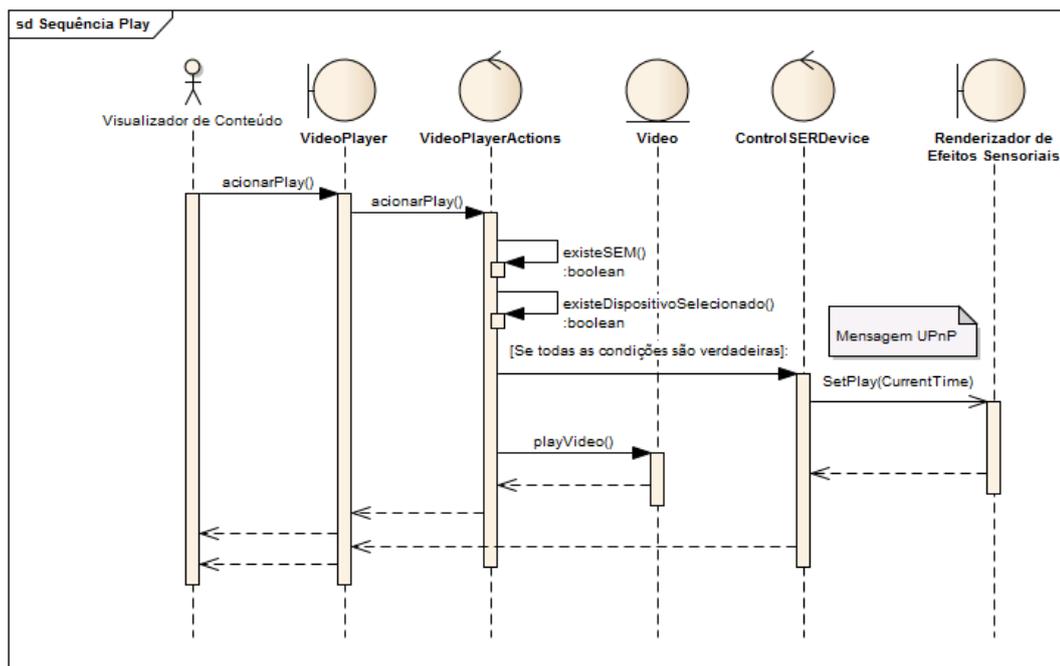


Fonte: Elaborada pelo autor

Na sequência “Pause”, a mensagem “acionarPlay” deve ser substituída por “acionar-Pause”, a mensagem “playVideo” substituída por “pauseVideo” e a mensagem assíncrona “SetPlay(CurrentTime)” substituída por “SetPause(CurrentTime)”. Na sequência “Stop”, a mensagem “acionarStop” deve ser substituída por “acionarStop”, a mensagem “playVideo” substituída por “stopVideo” e a mensagem assíncrona “SetPlay(CurrentTime)” substituída por “SetStop(CurrentTime)”. Na sequência “Posicionar vídeo no tempo”, a mensagem “acionarPlay” deve ser substituída por “posicionarVideoNoTempo”, a mensagem “playVideo” substituída por “posicionarVideoNoTempo” e a mensagem assíncrona “SetPlay(CurrentTime)” substituída por “SetCurrentTime(CurrentTime)”.

Para que as adaptações sejam realizadas efetivamente, é imprescindível perscrutar os diagramas de sequência apresentados nessa seção, observando principalmente a troca de mensagens UPnP, onde é realizada a comunicação entre o *Media Player* e o *SE Renderer*.

Figura 42 – Diagrama de sequência – Play.



Fonte: Elaborada pelo autor

4.4 Conclusão

Esse capítulo apresentou aspectos relacionados à engenharia de software da plataforma PlaySEM, levando em conta requisitos de reutilização, mobilidade e interoperabilidade. A abordagem proposta do renderizador de efeitos sensoriais desacoplado dos *Media Players* torna a solução capaz de ser reaproveitada em qualquer *Media Player*.

Os componentes utilizados na sua arquitetura permitem a abstração de complexidades inerentes à comunicação UPnP, processamento de vídeo e empacotamento e desempacotamento de SEM em objetos. O Cling UPnP Core é uma biblioteca Java que implementa essa infraestrutura, oferecendo APIs para desenvolvedores que precisam implementar serviços UPnP e pontos de controle. O VLCJ permite o desenvolvimento de *Media Players* em Java, abstraindo uma série de complexidades ao lidar com decodificação em diferentes formatos. A biblioteca MPEG *Metadata* facilita a ligação entre o XML (suportando todas as partes do MPEG-V) e objetos da aplicação.

Foram apresentadas as funcionalidades do PlaySEM SE *Video Player* e do SE *Renderer*, incluindo a descrição de serviços fornecidos pelo último. A parte de hardware do SE *Renderer* também foi abordada, descrevendo os materiais usados na construção dos atuadores e suas limitações. Por fim, foram apresentando diagramas comportamentais para adaptação de outros *Media Players* ao SE *Renderer*. Um vídeo contendo uma demonstração do PlaySEM *Video Player* funcionando em conjunto do SE *Renderer* está disponível em <http://youtu.be/aka4jBZcoKk>.

Trabalhos futuros na direção da melhoria da qualidade dos hardwares precisam ser realizados, tentando eliminar as limitações atuais. O suporte a diferentes atuadores já existentes de uso doméstico, como o amBX Gaming PC peripherals, Cyborg Gaming Lights, Vortex Activ, também podem ser realizados no SE *Renderer*, conferindo alternativas ao usuário que está consumindo o produto. Outras oportunidades serão discutidas no capítulo 6.

5 Experimentos e resultados

No capítulo anterior foi apresentada a plataforma PlaySEM sob o ponto de vista de arquitetura do software, onde foi possível detalhar principalmente o comportamento interno do SE *Renderer*. Contudo, é importante compreender se a abordagem do SE *Renderer* desacoplado tem efeito no processamento dos comandos enviados pelo programa cliente e verificar se a arquitetura proposta tem impacto relevante na dessincronização entre os relógios quando usado em linha de tempo. Além disso, é também relevante observar se o SE *Renderer* suporta a combinação com aplicações baseadas em eventos.

A seguir são descritos os métodos e materiais utilizados para confirmar ou invalidar hipóteses formuladas sobre a plataforma considerando o uso desacoplado das aplicações e suporte ao padrão MPEG-V.

5.1 Metodologia

O método científico usado para avaliar os resultados é hipotético-dedutivo de Popper. A Figura 43 adaptada da obra de Lakatos e Marconi (2003) descreve o processo. A fase inicial de expectativas e conhecimento prévio de teorias existentes dá origem ao problema. Diante do problema são constituídas soluções propostas com suposições ou hipóteses (conjecturas). Em seguida, as conjecturas (passíveis de refutação) passam por testes de falseamento por meio de observação e experimentos.

Figura 43 – Etapas do método hipotético-dedutivo.



Fonte: Adaptada de Lakatos e Marconi (2003)

Como discutido no capítulo 3, não foram encontradas na literatura abordagens de renderizadores de efeitos sensoriais desacoplados de *media players*, exigindo que o usuário visualizador de conteúdo utilize *players* com renderizadores embutidos restritos a ambientes específicos. Notadamente a solução de renderização dos efeitos sensoriais, cuja responsabilidade é interpretar e processar elementos que descrevem efeitos sensoriais, não é passível de reutilização nesse cenário. Uma nova abordagem é proposta nesse trabalho, onde o renderizador de efeitos sensoriais está desacoplado do *player* oferecendo serviços (com suporte ao padrão MPEG-V) em rede, não somente para *players* baseado em linha de tempo, mas para qualquer tipo de aplicação que necessite produzir efeitos sensoriais

sem que seja necessário conhecer o baixo nível de comunicação com os atuadores. Dessa proposta surgiram as seguintes hipóteses:

- i O fato de se comunicar em rede local não introduz atrasos relevantes para a reprodução dos efeitos sensoriais no ambiente do usuário;
- ii A arquitetura do SE *Renderer* tem impacto pequeno com relação à perda de sincronização entre efeitos sensoriais x *media players*;
- iii A abordagem desacoplada do SE *Renderer* dá suporte às aplicações MPEG-V com autoria baseada em eventos e linha de tempo.

Para testar cada hipótese foi realizada um planejamento do esboço das provas por meio de medições e observações, seguido da execução das operações planejadas e coleta os dados, elaboração de dados para classificação e análise, e por fim os dados foram interpretados.

Com o intuito de serem passíveis de reprodução, abaixo são listados os materiais para testes em cima da plataforma e na sequência são apresentados os métodos de execução.

- **Mini-PC** com o processador Intel Atom CPU 330 com 2GB de memória RAM, disco rígido de 80GB, adaptador de rede *wireless* Atheros AR9285 e conectado a microcontrolador Arduino na porta USB, rodando o sistema operacional Windows 7 32 bits, para a execução do SE *Renderer*;
- **Ultrabook** com o processador Intel Core I7-3537U com 4GB de memória RAM, disco SSD de 128 GB e adaptador *wireless* Intel Centrino Advanced-N 6235, rodando o sistema operacional Windows 8.1 64 bits, para execução do PlaySEM *Video Player* e da aplicação baseada em evento;
- **Access Point Wireless** D-link modelo WBR-1310 com suporte aos padrões 802.11g e 802.11b ligado em uma rede doméstica suportando a conexão de 1 *Smart TV*, 2 *smartphones* e 1 computador *desktop*, além do Mini-PC e do Ultrabook utilizados nos testes;
- **Sensor Kinect**, para utilização na aplicação interativa baseada em eventos;
- **Software WireShark** versão 1.12.4 configurado com o filtro $(ip.src == 90.0.0.106 \ \&\& \ ip.dst == 90.0.0.100) \ || \ (ip.src == 90.0.0.100 \ \&\& \ ip.dst == 90.0.0.106) \ \&\& \ (http.response \ || \ http.request \ || \ http.notification)$, para coleta de informações de rede no teste da hipótese (*i*);

- **Trailer do filme Indiana Jones** com duração de 1 minuto e 42 segundos anotado com auto-extração de luz do vídeo, 16 efeitos de vento e 7 efeitos de vibração descritos com o padrão MPEG-V, utilizado nos cenários de teste das hipóteses (i) e (ii).

O teste da primeira hipótese consiste em verificar se as mensagens trafegadas na rede para execução dos serviços disponibilizados pelo SE *Renderer* introduzem atrasos relevantes para a reprodução dos efeitos sensoriais no ambiente do usuário. A latência foi medida capturando pacotes *http* trafegados na rede, com início na requisição e término na resposta. A aplicação cliente usada, o *PlaySEM Video Player*, possui um parâmetro chamado de *stats*, se ele for verdadeiro são exibidas no console mensagens contendo $T1$ - tempo antes da execução do comando, $T2$ - tempo ao término de execução do comando e $T3$ - tempo de confirmação do processamento pelo SE *Renderer*. No caso do comando “SetSem” há uma particularidade para determinar o tempo de processamento, pois ele depende de uma notificação de evento do SE *Renderer* confirmando que o SEM foi processado e está pronto para ser executado. Nesse caso é exibido $T4$ - tempo de confirmação do processamento do SEM. Subtraindo $T3$ ($T4$ para *SetSem*) de $T1$ é possível obter o tempo de execução. Descontando o tempo de latência do tempo de execução, obtém-se o tempo de processamento local no SE *Renderer*.

No teste da segunda hipótese foram coletados tempos decorridos do relógio do *PlaySEM Video Player* e do SE *Renderer*. A aplicação *Video Player* é baseada em linha de tempo do vídeo e a precisão do relógio do componente VLC gira em torno de 300 milissegundos. Para melhorar a condição de medição do relógio da aplicação, foi criado um outro relógio com precisão de 1 milissegundo paralelo à linha principal da mídia principal. Com isso, é possível comparar os relógios internos das 2 aplicações de modo mais preciso, considerando que o SE *Renderer* também tem a precisão de 1 milissegundo. No *Video Player* há uma funcionalidade que exibe o tempo decorrido do relógio paralelo no console e é acionado pressionando a tecla “T” na sua interface. O SE *Renderer*, quando habilitado o parâmetro *debug*, o tempo do relógio e os comandos que foram processados são exibidos no console. Para confrontar os relógios foi executado o comando “SetPause” durante 5 rodadas. O valor de dessincronização foi obtido a partir da diferença entre o tempo do relógio do SE *Renderer* e do *Video Player*.

Para a terceira hipótese foi utilizado um estudo de caso que permite avaliar o comportamento do renderizador de efeitos sensoriais conectado a uma aplicação baseada em eventos. O cenário compreende uma aplicação interativa com reconhecimento de gestos e controle de contexto, em que são disparados efeitos luminosos que mudam o estado do ambiente. A ideia é que o desenvolvedor da aplicação interativa concentre esforços no seu domínio, deixando a cargo do SE *Renderer* o processamento e renderização dos efeitos sensoriais. Nesse estudo foram coletados os tempos de execução de 8 ações (descritas com MPEG-V) por meio de 2 comandos necessários para o funcionamento com esse tipo de

aplicação, o “SetSem” (pré-processamento) seguido do “SetPlay” (executado somente após a notificação de processamento do SEM). O método de coleta é semelhante ao processo de teste da primeira hipótese. Foi definida uma marca de tempo antes do envio do comando “SetSem”, outra após a notificação de evento do SE *Renderer* confirmando que o SEM foi processado e por último a confirmação de sucesso na execução do serviço “SetPlay” enviado pelo SE *Renderer*. Os tempos coletados servem de referência para o desenvolvedor avaliar se a solução é plausível ao contexto que ele deseja aplicar.

As evidências dos testes realizados estão disponíveis em <<https://goo.gl/kcmQ44>>.

5.2 Resultados da avaliação

5.2.1 Uso em rede

Nielsen (1993) afirma¹ em seu livro sobre engenharia de usabilidade que 100 milissegundos é o tempo limite para o usuário sentir que o sistema está reagindo instantaneamente e 1000 milissegundos para o fluxo de pensamento do usuário permanecer ininterrupto mesmo que ele note o atraso. Um estudo recente de Yuan et al. (2015), sobre os limites temporais tolerados entre a sincronização da mídia de vídeo com efeitos táteis e de fluxo de ar, indica que efeitos táteis podem sofrer um atraso de até 1s atrás da mídia principal, enquanto o fluxo de ar pode ser liberado tanto 5s à frente ou 3s atrás. Não são relatados outros tipos de efeitos.

No teste para verificar se as mensagens trafegadas na rede para execução dos serviços disponibilizados pelo SE *Renderer* introduzem atrasos relevantes para a reprodução dos efeitos sensoriais no ambiente do usuário, foram coletados dados de 5 rodadas e calculado o tempo médio. A Tabela 1 mostra os dados coletados para análise desse teste. A *LM* indica a Latência Média do tráfego na mensagem na rede, o σ (desvio padrão) indica a variação do tempo na *LM* para projetar quão maior ela poderia ser do ponto de vista estatístico. O *CV* indica em percentual o Coeficiente de Variação, projetado em *LMI*, Latência Mínima Média, para menos e em *LMA*, Latência Máxima Média, para mais. O *TMP* indica o Tempo Médio de Processamento Local no SE *Renderer*, descontando o tempo de latência do tempo de execução. O *TME* indica o Tempo Médio de Execução para cada um dos serviços de comando do SE *Renderer*.

Nos comandos com funções básicas de controle (*Play*, *Pause* e *Stop*) o tempo médio de atraso do tráfego na rede foi menor que 16 milissegundos, que decerto não acarreta prejuízo no tempo de execução do serviço. O encaminhamento do conjunto de cores extraídas do *frame* do vídeo por meio do serviço “SetLightColors” para efeitos de luz, que provavelmente um atraso seria mais visível aos olhos do usuário, também

¹ Reiterado na web no ano de 2014 e disponível em: <<http://www.nngroup.com/articles/response-times-3-important-limits/>>

Tabela 1 – Tempo médio de latência na execução dos serviços do SE *Renderer*.

Serviço	LM (ms)	σ (ms)	CV (%)	LMI (ms)	LMA (ms)	TMP (ms)	TME (ms)
SetSem	51	11,58	22,71	39,42	62,58	280	331
SetPlay	12	1,14	9,50	10,86	13,14	58	70
SetPause	13	2,5	19,23	10,50	15,50	17	30
SetCurrentTime	24	20,37	84,88	3,63	44,37	21	45
SetLightColors	15	2,96	19,73	12,04	17,96	10	25
SetStop	16	1,34	8,38	14,66	17,34	11	27

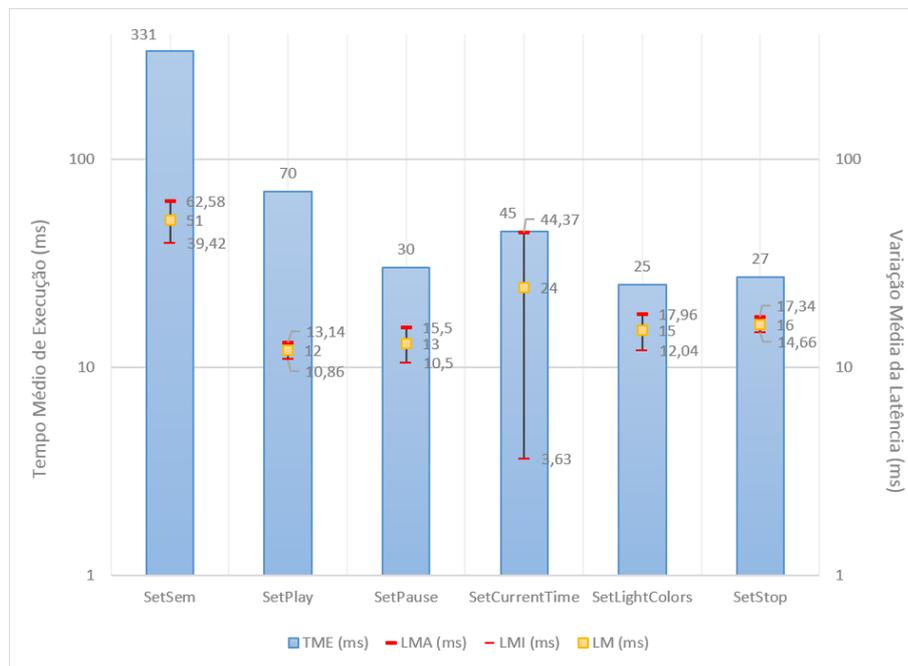
Fonte: Elaborada pelo autor

apresenta uma latência média minimalista. Se a aplicação seguir a abordagem tradicional (renderizador embutido no *Video Player*), a eliminação do atraso provocado pela rede frente ao *TME* não traria ganho expressivo. A alta dispersão apresentada pelo *CV* do serviço “SetCurrentTime” provavelmente se justifica pela inundação de solicitações do *player* no SE *Renderer*, ocasionado por eventos originados na interface da aplicação durante o posicionamento do tempo com o mouse no componente *JSlider*, que representa a linha de tempo da mídia.

A figura 44 apresenta os dados de *TME* em relação à *LM* considerando as possíveis variações a partir da amostra coletada e evidencia a afirmação sobre a alta dispersão dos dados no serviço “SetCurrentTime”. Em algum momento é bem provável que o tempo de atraso de aproxime do tempo de execução, contudo, considerando os parâmetros fornecidos no início dessa seção, a máxima está dentro do limite aceitável para aplicações multimídia com efeitos sensoriais.

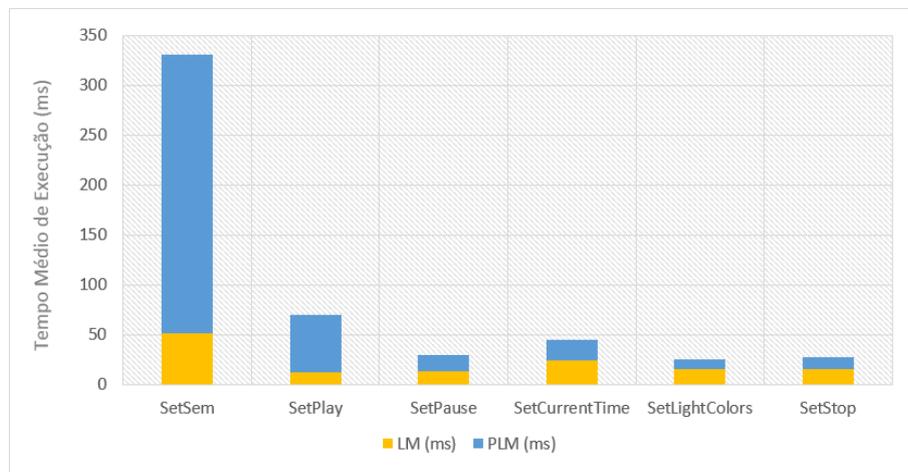
Na Figura 45 nota-se que a *LM* de todos os serviços está quase sempre abaixo de 25 milissegundos, exceto quando carrega uma quantidade maior de informação, como o caso do serviço “SetSem”, onde são trafegados os dados da anotação de todos os efeitos sensoriais da mídia principal.

Figura 44 – Tempo de execução dos comandos x Variação da latência.



Fonte: Elaborada pelo autor

Figura 45 – Latência em relação ao tempo total de processamento.



Fonte: Elaborada pelo autor

5.2.2 Impacto na perda de sincronização

Os resultados dos testes para verificar se arquitetura com o SE *Renderer* teve baixo impacto em relação à perda de sincronização entre o *player* e os efeitos sensoriais são apresentados na tabela 2. Foram medidos dois relógios: *R1*, relativo ao *Video Player*, e *R2*, relativo ao SE *Renderer*. Aproximadamente a cada 3 segundos um comando “SetPause” foi disparado a partir do *Video Player* para o SE *Renderer* com o intuito de “congelar” os relógios de ambas aplicações. Os tempos foram coletados em 5 rodadas denominadas de

$P1$, $P2$, $P3$, $P4$ e $P5$.

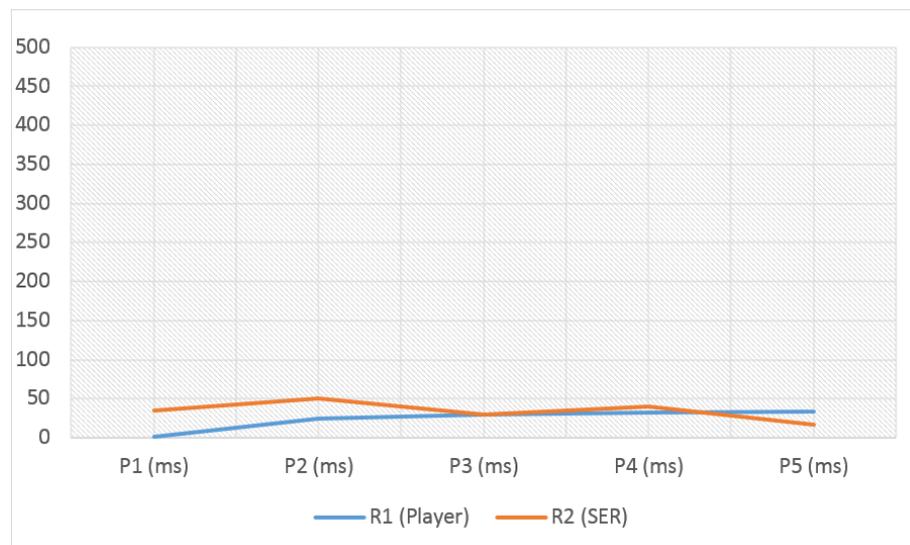
Tabela 2 – Diferença de tempo entre o relógio do *Player* e do SE *Renderer* em milissegundos.

Variável	P1 (ms)	P2 (ms)	P3 (ms)	P4 (ms)	P5 (ms)
R1 (<i>Player</i>)	3001	5825	8830	11832	14834
R2 (SE <i>Renderer</i>)	3035	5851	8830	11840	14817
Diferença	34	26	0	8	-17

Fonte: Elaborada pelo autor

A Figura 46 exibe o gráfico contendo a diferença de tempo entre o relógio do *Player* e do SE *Renderer* em milissegundos. O relógio $R2$ adiantado 17 milissegundos em $P5$ é justificado pelo atraso no processamento de retomada do vídeo em $P4$. Provavelmente o tempo da requisição assíncrona “SetPlay” chegou ao SE *Renderer* antes que o componente VLC renderizasse o vídeo.

Figura 46 – Diferença de tempo entre o relógio do *Player* e do SE *Renderer* em milissegundos.



Fonte: Elaborada pelo autor

Se o usuário visualizador do conteúdo multimídia pode suportar a diferença relatada por Yuan et al. (2015) de forma razoável, a perda de sincronização entre o *player* e os efeitos sensoriais relatados no PlaySEM pode ser considerada irrelevante, embora não contabilize o tempo que o hardware esteja efetivamente em funcionamento.

5.2.3 Estudo de caso: uso em aplicação interativa baseada em evento

Esse estudo de caso foi utilizado para verificar se a abordagem desacoplada do SE *Renderer* dá suporte às aplicações MPEG-V baseadas em eventos, além da linha de tempo

como as demais soluções. Trata-se de um ambiente interativo que integra um *framework* de reconhecimento de gestos com efeitos sensoriais descritos com o MPEG-V (SANTOS; NETO; SALEME, 2015).

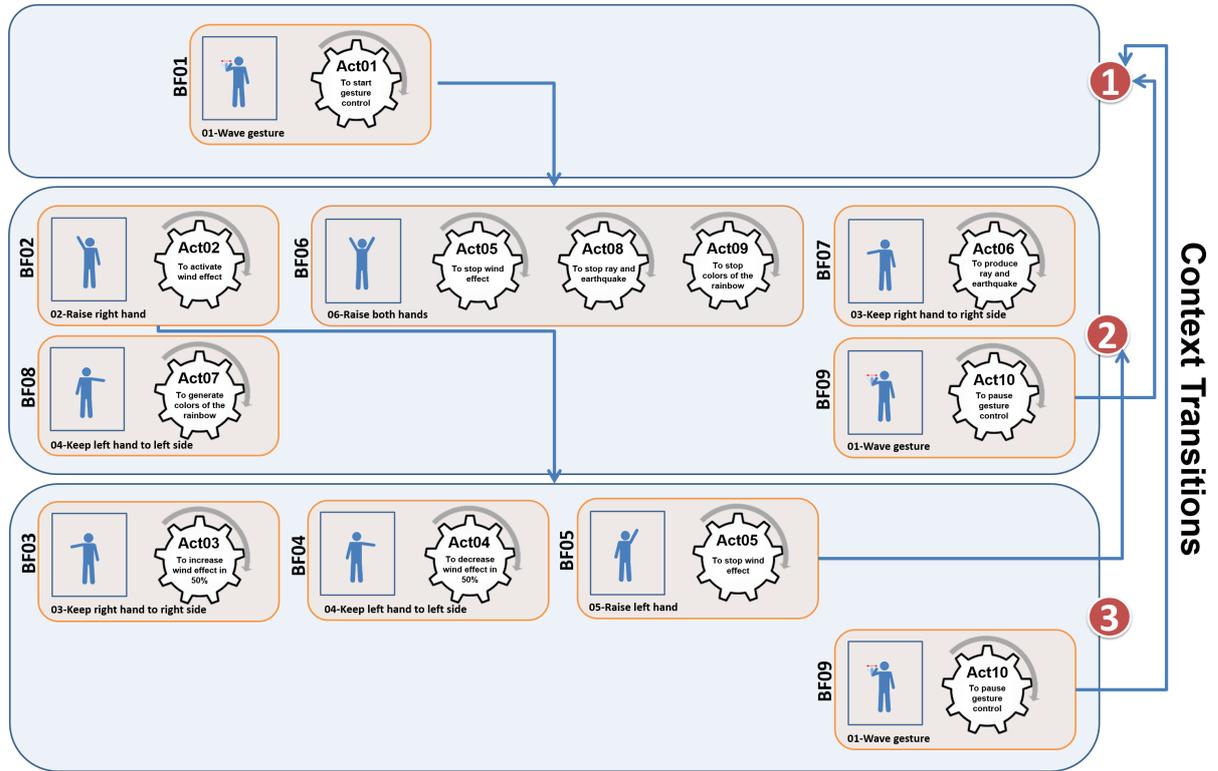
O cenário previsto inclui um quarto com uma interface de captura de gesto, um computador para processar a informação desse sensor e controlar o envio das mensagens compatíveis com MPEG-V para o SE *Renderer* para ativar, desativar ou alterar o estado dos atuadores distribuídos por toda a sala. Os componentes que constituem o ambiente utilizado no estudo de caso são (i) um sensor Kinect como dispositivo de entrada para o reconhecimento de gesto; (ii) um PC para controlar a execução ambiente; (iii) um Mini-PC executando o SE *Renderer*, para receber efeitos sensoriais descritos com MPEG-V e para produzir uma saída para os atuadores do mundo real por meio de um microcontrolador Arduino; (iv) um monitor, apresentando objetos virtuais; e (v) uma fita de LED, para efeito de iluminação. Apesar da ausência dos atuadores para efeito de vento e vibração nesse experimento, os comandos para esses dispositivos são processados normalmente pelo SE *Renderer*.

A aplicação implementa os conceitos do *framework* TTAir (NETO; SANTOS; CARVALHO, 2013) e foi desenvolvida e escrita na linguagem C# .NET. Além do reconhecimento de gestos, ele fornece um conjunto de camadas em sua arquitetura para simplificar o desenvolvimento de aplicações interativas e suportar o desenvolvimento por etapas. Como entrada para a aplicação, foi mapeado um XML contendo o conjunto de interações possíveis em cada contexto e ações associadas (ver Figura 47), de forma que o *framework* controle as transições automaticamente. Por exemplo, se o controle de gestos não foi acionado, então não é possível realizar outras interações.

Ao iniciar a aplicação, para acionar o controle por gestos o usuário que está interagindo deve realizar o gesto “01-Wave gesture” que acionará a ação “Act01-To start gesture control” para ativar o controle por gestos. Em seguida, ocorre a transição de estado para o segundo contexto, que disponibiliza outros gestos que estão associados às ações contendo efeitos sensoriais descritos no padrão MPEG-V. Por exemplo, ao executar o gesto “04-Keep left hand to left side” a aplicação dispara a ação “Act07-To generate colors of the rainbow” que na prática aciona o serviço “SetSem” seguido de “SetPlay” no SE *Renderer* para respectivamente definir os metadados com as descrições dos efeitos sensoriais e reproduzi-los no renderizador. A Figura 48 mostra o esboço do ambiente e uma sequência de efeitos luminosos no ambiente para a ação exemplificada. Um vídeo com a demonstração está disponível em: <<http://youtu.be/a5QPsB2zKpk>>.

A Tabela 3 apresenta tempos de referência para cada uma das ações do estudo de caso. São apresentadas as quantidades de efeitos em cada delas e os tempos *TMPP* - Tempo Médio de Pré-processamento e *TMT* = Tempo Médio Total que indicam respectivamente o tempo médio de processamento do SEM e o tempo médio total entre a solicitação de

Figura 47 – Conjunto de interações e ações especificadas em cada contexto.



Fonte: (SANTOS; NETO; SALEME, 2015)

Figura 48 – À esquerda um esboço do ambiente e seus componentes. À direita é mostrado a sequência de efeitos no ambiente real.



Fonte: Elaborada pelo autor

execução da ação e sua finalização. A média foi obtida a partir da execução de 5 rodadas de teste.

A partir dos dados da Tabela 3 nota-se que os tempos não são determinísticos no cenário com pouca variação de números de efeitos sensoriais e as variações de tempo são praticamente imperceptíveis, com diferença de 73 milissegundos entre o maior e menor *TM*. Quando o número de efeitos sensoriais é aumentado significativamente, conforme mostrado na Tabela 4, há um acréscimo esperado no *TMPP*. O resultado destoante da ação ACT03 provavelmente foi ocasionado devido a oscilação não frequente na rede em

Tabela 3 – Tempo de pré-processamento dos *scripts* MPEG-V usados no estudo de caso.

Ação	Efeito de luz	Efeitos de vento	Efeitos de vibração	TMPP (ms)	TMT (ms)
ACT02	0	1	0	180	210
ACT03	0	1	0	245	274
ACT04	0	1	0	195	222
ACT05	0	1	0	195	227
ACT06	5	0	2	180	207
ACT07	8	0	0	195	221
ACT08	1	0	0	173	201
ACT09	0	0	1	171	201

Fonte: Elaborada pelo autor

um determinado instante, elevando o *TMPP* e o *TM*.

Tabela 4 – Tempo de pré-processamento de *script* MPEG-V ampliado com 90 efeitos sensoriais para aplicação orientada a evento.

Ação	Efeito de luz	Efeitos de vento	Efeitos de vibração	TMPP (ms)	TMT (ms)
ACTXX	30	30	30	287	317

Fonte: Elaborada pelo autor

Além da abordagem baseada em eventos, esse estudo de caso também se diferencia dos experimentos anteriores apresentados nessa seção por usar ambiente de programação diferente do *Video Player* para se comunicar com o SE *Renderer*, fazendo uso da biblioteca *ManagedUPnP*². A partir dos dados coletados, percebeu-se que a mudança de ambiente de desenvolvimento praticamente não teve impacto no tempo de comunicação com o SE *Renderer*. Para efeito de comparação, a soma dos tempos *TME* dos serviços “SetSem” e “SetPlay” exibidos na Tabela 1 resultam no valor de 401 milissegundos, uma diferença de 84 milissegundos a mais que o *TMT* mostrado na Tabela 4, com 67 efeitos a menos no total.

A conclusão sobre os valores obtidos na avaliação será apresentada no próximo capítulo, juntamente com as discussões finais desse trabalho.

² Biblioteca *ManagedUPnP* disponível em <<https://managedupnp.codeplex.com/>>

6 Conclusão e trabalhos futuros

Como contribuição principal essa dissertação propõe e implementa uma nova abordagem para renderização de efeitos sensoriais de modo desacoplado dos reprodutores tradicionais de multimídia, permitindo a sua reutilização por outras soluções. O renderizador suporta o padrão MPEG-V, que inclui a especificação de representações sobre efeitos sensoriais. Apesar de definir as condições para especificação, o valor real acrescentado está nas aplicações que transformam dados brutos em algo essencialmente útil.

A questão da interoperabilidade com o MPEG-V permite que mídias de diferentes produtores de conteúdo possam ser lidas e processadas pelo renderizador de efeitos sensoriais. A plataforma PlaySEM está fundamentada principalmente na parte 3 e também aspectos de informações sobre dispositivos relatados na parte 2. Naturalmente, faz uso da parte 6, que é utilizada como apoio das outras partes.

As ferramentas de autoria de efeitos sensoriais, *Media Players* e atuadores para renderizadores de efeitos sensoriais atualmente abordadas pela literatura foram apresentadas e discutidas. Além da abordagem desacoplada do renderizador, outras características que diferem essa solução das demais são o baixo custo do hardware associado, por reaproveitar componentes eletrônicos de outros dispositivos legados, o suporte multi-plataforma, permitindo que tanto o SE *Video Player* quanto o SE *Renderer* sejam executados em sistemas operacionais diversos como Windows e Linux, e a flexibilidade na substituição de componentes aproveitando-se da sua arquitetura.

A arquitetura da solução (SE *Video Player* e SE *Renderer*) se baseia em componentes abertos, como Cling UPnP Core para comunicação UPnP entre o *Video Player* e o renderizador, VLCJ para reproduzir vídeos a partir da própria interface do programa Java, e o MPEG *Metadata* para empacotar e desempacotar XML contendo metadados de efeitos sensoriais. Ela foi projetada para permitir que o renderizador de efeitos sensoriais bem como os atuadores possam ser reaproveitados em outras interfaces de apresentação, não se limitando a aplicações baseadas em linha de tempo. Diagramas estruturais e comportamentais foram apresentados para facilitar a compreensão da construção do conjunto de softwares e também adaptar outros *Media Player's* para reutilização do renderizador de efeitos sensoriais.

Com relação aos hardwares, há algumas ressalvas. Uma está relacionada com a reprodução de efeitos de vento e vibração em intensidade abaixo de 30%. As intensidades são reguladas por meio de tensão e abaixo desse percentual os dispositivos não produzem os efeitos por falta de potência. Para contornar esse problema, toda vez que os dispositivos são acionados até a faixa de 30% ele automaticamente assume esse valor. Um outro

inconveniente pode ser o sinal sonoro emitido pelos ventiladores em tensões reduzidas, o que não se caracteriza como uma limitação.

O método para validar a solução constituiu-se de conjecturas formadas principalmente em função da abordagem desacoplada, sendo: (i) o fato de se comunicar em rede local não introduz atrasos relevantes para a reprodução dos efeitos sensoriais no ambiente do usuário; (ii) a arquitetura do SE *Renderer* tem impacto pequeno com relação à perda de sincronização entre efeitos sensoriais x *media players*; (iii) a abordagem desacoplada do SE *Renderer* dá suporte às aplicações MPEG-V com autoria baseada em eventos e linha de tempo. Para analisá-las, dados de diferentes cenários foram coletados e discutidos. Confrontando os limites de tempo de resposta encontrados na literatura acerca de aplicações multimídia, pode-se tomar como verdade as hipóteses levantadas.

Nos comandos com funções básicas de controle (*Play, Pause e Stop*) o tempo médio de atraso do tráfego na rede foi menor que 16 milissegundos. Se a aplicação seguir a abordagem tradicional (renderizador embutido no *Video Player*), a eliminação do atraso provocado pela rede frente ao tempo médio de execução não traria ganho expressivo. A latência média de todos os serviços foi quase sempre abaixo de 25 milissegundos, exceto quando houve uma quantidade maior de informação sendo trafegada, aplicável ao serviço de carregar os metadados de efeitos sensoriais. Nesse caso, a latência média foi de 51 milissegundos.

Com relação à perda de sincronização entre efeitos sensoriais x *media players*, notou-se períodos em que o vídeo esteve na frente ou atrás, mas com diferença máxima de 34 milissegundos. A partir do relato das experiências encontradas na literatura, sobre os limites temporais tolerados entre a sincronização da mídia de vídeo com efeitos táteis (1s atrás) e de fluxo de ar (5s à frente ou 3s atrás), a perda de sincronização entre o *player* e os efeitos sensoriais relatados no PlaySEM pode ser considerada irrelevante, embora não contabilize o tempo que o hardware esteja efetivamente em funcionamento.

Uma dificuldade decorrente do modelo atual de renderização de efeitos sensoriais é a indução da entrada das descrições MPEG-V nos *Media Players* por meio de linha de tempo (paradigma *timeline*), diferente dos modelos geralmente desenvolvidos para aplicações baseadas em eventos. O caso de estudo revelou a possibilidade para desenvolver e integrar aplicativos com interação natural e disparar ações associadas com efeitos sensoriais apenas especificando *scripts* com o padrão MPEG-V no programa cliente e transmitindo para o SE *Renderer* no tempo desejado, sem a necessidade de conhecer os hardwares e como programá-los. O estudo permitiu evidenciar a versatilidade do renderizador de efeitos sensoriais em tipos de aplicação distintas de linha de tempo.

6.1 Trabalhos futuros

Os resultados obtidos com o desenvolvimento dessa dissertação podem ser ampliados e reutilizados em outros trabalhos. A começar por evoluções no conjunto de hardwares para melhorar a qualidade dos atuadores mantendo o baixo custo. Adicionalmente, pode ser incluído o suporte de outros tipos de efeitos sensoriais, como aroma e névoa, e também suporte a diferentes atuadores já existentes de uso doméstico, como o amBX Gaming PC peripherals, Cyborg Gaming Lights, Vortex Activ, dentro outros. Para suportar a execução do renderizador de efeitos sensoriais, hardwares com característica mais ubíqua podem ser utilizados.

Com relação ao SE *Video Player*, para melhorar a auto-extração de cores do vídeo o intervalo de captura de *frames* deve ser reduzido com foco em prover ao renderizador mais informações em menos tempo. Além disso, o desenvolvimento de outros *Video Player's* compatíveis com o renderizador precisam ser desenvolvidos para outras plataformas, destacando principalmente *Smart TV's*. A questão da segurança do dispositivo na rede deve ser trabalhada, para evitar o acesso indiscriminado ao renderizador, que poderia gerar efeitos não desejados. A adaptação da plataforma às preferências do usuário também pode ser introduzida, utilizando a parte 2 do padrão MPEG-V. Por exemplo, se o usuário não deseja o efeito de vento por algum motivo particular, o renderizador deve ignorar o processamento desse tipo de efeito.

Verificou-se que a abordagem desacoplada entre o *player* e o renderizador de efeitos sensoriais não impacta de forma significativa no atraso da renderização dos efeitos sensoriais. No entanto, é importante que outros trabalhos avaliem o tempo para acionar efetivamente o hardware. No cenário de efeitos sensoriais, a tarefa de projetar e orquestrar eficientemente a comunicação entre softwares e hardwares não é trivial. Conhecendo mais especificamente os requisitos desse tipo de aplicação, as soluções poderiam ser projetadas considerando a acurácia necessária.

No campo de QoE é importante que sejam desenvolvidos trabalhos no sentido de capturar a sensação do usuário de modo imparcial, para aumentar a precisão sobre o impacto de cada efeito na cena para o usuário. A introdução de avaliações com o uso de EEG (Eletroencefalograma) apontam para esse caminho. Outras formas de captura das sensações poderiam ser investigadas, como o uso de almofadas ou colchões com sensores, permitindo a captura e o processamento das sensações em tempo real, para que o software tome decisões em cima da percepção com o intuito de melhorar a sensação momentânea do usuário. Essas informações de captura sobre as sensações do usuário poderiam ser especificadas no próprio padrão MPEG-V, desde que ele fosse estendido para isso.

Por último, talvez o maior desafio seja a geração automática de metadados de efeitos sensoriais em MPEG-V a partir do processamento audiovisual para extrair semântica

das ações. É um problema semelhante à recuperação de conteúdo audiovisual, com o desafio adicional de gerar uma saída com metadados que representem fielmente as cenas, considerando atributos como intensidade e intervalo de tempo. O emprego de técnicas de inteligência artificial, reconhecimento de padrões e visão computacional pode ser o caminho de novas pesquisas na área de múltiplos efeitos sensoriais.

Referências

- CAMBRIDGE. *Render - definition in the American English Dictionary*. 2015. Disponível em: <<http://dictionary.cambridge.org/us/dictionary/american-english/render>>. Acesso em: 20.2.2015. Citado na página 21.
- CAPRICA. *Java framework for the VLC media player*. 2015. Disponível em: <<https://github.com/caprica/vlcj>>. Acesso em: 24.2.2015. Citado na página 65.
- CHO, H.-Y. *Event-Based control of 4D effects using MPEG RoSE*. 2010. Citado 7 vezes nas páginas 22, 45, 51, 53, 56, 58 e 59.
- CHOI, B.; LEE, E.-S.; YOON, K. Streaming media with sensory effect. In: *Information Science and Applications (ICISA), 2011 International Conference on*. [S.l.: s.n.], 2011. p. 1–6. Citado 6 vezes nas páginas 45, 47, 48, 58, 70 e 71.
- CHOI, B. S.; KIM, S. K. *Text of ISO/IEC FDIS 23005-3 2nd edition Sensory Information*. 2012. Citado 8 vezes nas páginas 22, 31, 32, 33, 34, 35, 36 e 70.
- CLING. *Cling - Java/Android UPnP library and tools*. 2015. Disponível em: <<http://4thline.org/projects/cling/>>. Acesso em: 24.2.2015. Citado na página 65.
- DALEAIR. *Interoperability*. 2015. Disponível em: <<http://www.daleair.com/vortex-activ>>. Acesso em: 8.3.2015. Citado na página 55.
- EBRAHIMI, T. Quality of multimedia experience: Past, present and future. In: *Proceedings of the 17th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2009. (MM '09), p. 3–4. ISBN 978-1-60558-608-3. Disponível em: <<http://doi.acm.org/10.1145/1631272.1631275>>. Citado na página 19.
- ECIS. *European Committee for Interoperable Systems - Interoperability*. 2015. Disponível em: <<http://www.ecis.eu/ecis-interoperability>>. Acesso em: 10.3.2015. Citado na página 22.
- GHINEA, G.; ANDRES, F.; GULLIVER, S. R. *Multiple Sensorial Media Advances and Applications: New Developments in MulSeMedia: New Developments in MulSeMedia*. Information Science Reference, 2011. (Premier reference source). ISBN 9781609608224. Disponível em: <<http://books.google.com.br/books?id=VkFFGq4MpNQC>>. Citado 2 vezes nas páginas 20 e 21.
- GHINEA, G. et al. Mulsemedia: State of the art, perspectives, and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 11, n. 1s, p. 17:1–17:23, out. 2014. ISSN 1551-6857. Disponível em: <<http://doi.acm.org/10.1145/2617994>>. Citado 3 vezes nas páginas 19, 20 e 21.
- HAN, J. J.; PREDA, M. *Text of ISO/IEC FDIS 23005-4 2nd edition Virtual World Object Characteristics*. 2012. Citado 5 vezes nas páginas 22, 36, 37, 38 e 39.
- HAN, J. J.; YOON, K. R. *Text of ISO/IEC FDIS 23005-2 2nd edition Control Information*. 2012. Citado 3 vezes nas páginas 22, 29 e 30.

- ISO/IEC/IEEE Systems and software engineering – Architecture description. *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, p. 1–46, Dec 2011. Citado na página 67.
- KIM, S.-K. Authoring multisensorial content. *Signal Processing: Image Communication*, v. 28, n. 2, p. 162 – 167, 2013. ISSN 0923-5965. MPEG-V. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0923596512001993>>. Citado 5 vezes nas páginas 45, 46, 47, 52 e 58.
- KIM, S. K.; HAN, J. J. *Text of white paper on MPEG-V*. 2014. Citado 10 vezes nas páginas 19, 27, 29, 31, 32, 36, 37, 38, 40 e 41.
- KIM, S. K.; HAN, J. J.; YOON, K. R. *Text of ISO/IEC FDIS 23005-5 2nd edition Data Formats for Interaction Device*. 2012. Citado 4 vezes nas páginas 22, 27, 28 e 40.
- KIM, S.-K.; JOO, Y. S. Sensible media simulation in an automobile application and human responses to sensory effects. *ETRI Journal*, ETRI, v. 35, n. 6, p. 1001–1010, dez. 2014. ISSN 1225-6463. Disponível em: <<http://dx.doi.org/10.4218/etrij.13.2013.0038>>. Citado 5 vezes nas páginas 21, 45, 52, 53 e 58.
- KIM, S. K.; JOO, Y. S.; CHOI, B. S. *Text of ISO/IEC FDIS 23005-7 2nd edition Conformance and reference software*. 2013. Citado 2 vezes nas páginas 22 e 43.
- LAKATOS, E. M.; MARCONI, M. d. A. *Fundamentos de metodologia científica*. São Paulo, Brasil: Atlas, 2003. ISBN 8522433976. Citado na página 91.
- LARMAN, C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004. ISBN 0131489062. Citado na página 85.
- MADCATZ. *Cyborg Gaming Lights for PC*. 2015. Disponível em: <<http://store.madcatz.com/brands/mad-catz/Cyborg-amBX-Gaming-Lights-for-PC.html>>. Acesso em: 8.3.2015. Citado 2 vezes nas páginas 54 e 55.
- MCROBERTS, M. *Arduino básico*. 1. ed. [S.l.]: Novatec Editora, 2011. 456 p. ISBN 978-85-7522-274-4. Citado na página 82.
- MÖLLER, S.; RAAKE, A. (Ed.). *Quality of Experience: Advanced Concepts, Applications and Methods*. Cham: Springer, 2014. ISBN 978-3-319-02680-0. Citado 3 vezes nas páginas 19, 20 e 21.
- NETO, A. N. R.; SANTOS, C. A. S.; CARVALHO, L. A. a. de. Touch the air: An event-driven framework for interactive environments. In: *Proceedings of the 19th Brazilian Symposium on Multimedia and the Web*. New York, NY, USA: ACM, 2013. (WebMedia '13), p. 73–80. ISBN 978-1-4503-2559-2. Disponível em: <<http://doi.acm.org/10.1145/2526188.2526216>>. Citado na página 98.
- NIELSEN, J. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 0125184050. Citado na página 94.
- ORACLE. *JavaFX: Incorporating Media Assets Into JavaFX Applications*. 2015. Disponível em: <<http://docs.oracle.com/javase/8/javafx/media-tutorial/overview.htm>>. Acesso em: 24.2.2015. Citado na página 65.

- ORACLE. *JAXB Architecture*. 2015. Disponível em: <<http://docs.oracle.com/javase/tutorial/jaxb/intro/arch.html>>. Acesso em: 24.2.2015. Citado na página 66.
- OXFORD. *Render - definition od render in Oxford dictionary*. 2015. Disponível em: <http://www.oxforddictionaries.com/definition/american_english/render>. Acesso em: 20.2.2015. Citado na página 21.
- PHILIPS. *Philips amBX Gaming PC peripherals SGC5103BD Premium kit*. 2015. Disponível em: <http://www.p4c.philips.com/cgi-bin/cpindex.pl?ctn=SGC5103BD/27&hlt=Link_Overview&scy=US&slg=AEN>. Acesso em: 8.3.2015. Citado 2 vezes nas páginas 53 e 54.
- PREDA, M. *MPEG-V: a standard for multissensorial and immersive experiences*. 2011. Disponível em: <<http://pt.slideshare.net/MariusPreda/mpeg-vawareness-event>>. Acesso em: 30.3.2015. Citado 2 vezes nas páginas 19 e 20.
- PRESSMAN, R.; MAXIM, B. *Software Engineering: A Practitioner's Approach*. 8. ed. [S.l.]: McGraw-Hill Science/Engineering/Math, 2014. ISBN 0078022126. Citado 5 vezes nas páginas 67, 68, 71, 74 e 85.
- RAINER, B. et al. Investigating the Impact of Sensory Effects on the Quality of Eexperience and Emotional Response in Web Videos. In: *4th International Workshop on Quality of Multimedia Eexperience (QoMEX'12)*. [s.n.], 2012. Disponível em: <http://www-itec.uni-klu.ac.at/bib/files/QoMEX2012_UserStudy.pdf>. Citado na página 19.
- SANTOS, C. A. S.; NETO, A. N. R.; SALEME, E. B. An Event Driven Approach for Integrating Multi-Sensory effects to Interactive Environments. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2015)*. [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 98 e 99.
- SANTOS, C. A. S.; SALEME, E. B.; ANDRADE, J. C. S. A systematic review of data exchange formats in advanced interaction environments. *IJMUE - International Journal of Multimedia and Ubiquitous Engineering*, SERSC, v. 10, n. 5, maio 2015. ISSN 1975-0080. Disponível em: <http://www.sersc.org/journals/IJMUE/vol10_no5_2015/13.pdf>. Citado na página 22.
- SOMMERVILLE, I. *Software engineering*. 9. ed. [S.l.]: Pearson, 2011. ISBN 9780137053469. Citado 4 vezes nas páginas 67, 68, 71 e 74.
- SUK, C. B.; HYUN, J. S.; YONG, L. H. Sensory effect metadata for smmd media service. In: PERRY, M. et al. (Ed.). *ICIW*. IEEE Computer Society, 2009. p. 649–654. Disponível em: <<http://dblp.uni-trier.de/db/conf/iciw/iciw2009.html#SukHY09>>. Citado 2 vezes nas páginas 70 e 71.
- UPNPFORUM. *UPnP Device Architecture 1.0*. 2008. Disponível em: <<http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf>>. Acesso em: 24.2.2015. Citado na página 64.
- VIDEOLAN. *VLC media player*. 2015. Disponível em: <<http://www.videolan.org/vlc/>>. Acesso em: 24.2.2015. Citado na página 65.

- WALTL, M. et al. An end-to-end tool chain for sensory experience based on mpeg-v. *Sig. Proc.: Image Comm.*, v. 28, n. 2, p. 136–150, 2013. Disponível em: <<http://dblp.uni-trier.de/db/journals/spic/spic28.html#WaltlIRTH13>>. Citado 11 vezes nas páginas 45, 46, 49, 50, 51, 53, 54, 55, 58, 59 e 67.
- WALTL, M.; TIMMERER, C.; HELLWAGNER, H. A Test-Bed for Quality of Multimedia Experience Evaluation of Sensory Effects. In: *First International Workshop on Quality of Multimedia Experience (QoMEX 2009)*. [s.n.], 2009. Disponível em: <http://www-itec.uni-klu.ac.at/bib/files/qomex2009_mwcthh.pdf>. Citado 4 vezes nas páginas 19, 20, 31 e 33.
- WALTL, M.; TIMMERER, C.; HELLWAGNER, H. Improving the Quality of Multimedia Experience through Sensory Effects. In: *2nd International Workshop on Quality of Multimedia Experience (QoMEX'10)*. [s.n.], 2010. Disponível em: <http://www-itec.uni-klu.ac.at/bib/files/qomex2010_mwcthh.pdf>. Citado na página 19.
- WITMER, B. G.; SINGER, M. J. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoper. Virtual Environ.*, MIT Press, Cambridge, MA, USA, v. 7, n. 3, p. 225–240, jun. 1998. ISSN 1054-7460. Disponível em: <<http://dx.doi.org/10.1162/105474698565686>>. Citado na página 20.
- YOON, K. R.; HAN, J. J. *Text of ISO/IEC FDIS 23005-6 2nd edition Common Data Format*. 2012. Citado 3 vezes nas páginas 22, 33 e 42.
- YUAN, Z. et al. Perceived synchronization of multimedia services. *Multimedia, IEEE Transactions on*, PP, n. 99, p. 1–1, 2015. ISSN 1520-9210. Citado 2 vezes nas páginas 94 e 97.
- YUAN, Z.; GHINEA, G.; MUNTEAN, G. Beyond multimedia adaptation: Quality of experience-aware multi-sensorial media delivery. *IEEE Transactions on Multimedia*, v. 17, n. 1, p. 104–117, 2015. Disponível em: <<http://dx.doi.org/10.1109/TMM.2014.2371240>>. Citado na página 19.