

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA**

Efeitos Sensoriais em NCL

Renato de Oliveira Rodrigues

Prof. Orientador:

Joel André Ferreira dos Santos, D.Sc.

**Rio de Janeiro,
Julho de 2019**

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA**

Efeitos Sensoriais em NCL

Renato de Oliveira Rodrigues

Projeto final apresentado em cumprimento às
normas do Departamento de Educação
Superior do Centro Federal de Educação
Tecnológica Celso Suckow da Fonseca,
CEFET/RJ, como parte dos requisitos para
obtenção do título de Bacharel em Ciência da
Computação.

Prof. Orientador:

Joel André Ferreira dos Santos, D.Sc.

**Rio de Janeiro,
Julho de 2019**

RESUMO

Aplicações multimídia se limitam a estimular dois sentidos humanos: visão e audição. Estudos recentes, procuram expandir a definição de aplicações multimídia, de forma a incluir estímulos para outros sentidos humanos. Dessa forma, incluem-se nas aplicações efeitos sensoriais que devem ser disparados de maneira sincronizada com o conteúdo audiovisual sendo apresentado. Com a inclusão de efeitos sensoriais, busca-se melhorar a qualidade de experiência do usuário com essas aplicações.

O padrão MPEG-V fornece uma arquitetura e forma de representação de informações para a comunicação entre a aplicação e os equipamentos responsáveis por executar os efeitos sensoriais. No entanto, é importante destacar que o padrão não define como a aplicação deve se comportar. O autor é responsável por especificar como deve ocorrer a apresentação do conteúdo audiovisual, sua sincronização, e a captura da interação com o usuário.

A fim de facilitar a autoria de aplicações multimídia, este trabalho propõe uma extensão do *middleware* Ginga e da linguagem Nested Context Language (NCL) de forma a prover uma abordagem declarativa para a criação destas aplicações e permitir aos autores reutilizar os recursos da linguagem para a sincronização de conteúdo, interação com o usuário, etc. Para isso, este trabalho irá estender a definição dos objetos de mídia da linguagem NCL criando um novo tipo de objetos de mídia de efeitos sensoriais. O *middleware* também será estendido de forma a definir um reprodutor de efeitos sensoriais que é responsável por controlar os dispositivos sensoriais durante a apresentação.

Palavras-chave: Multimídia; Efeitos Sensoriais; NCL; MPEG-V

ABSTRACT

Multimedia applications are limited to stimulate two human senses: sight and hearing. Recent studies seek to expand multimedia applications definition, in order to include stimuli to other human senses. Thus, such works include in applications sensory effects that are executed in sync with the audiovisual content being presented. By including sensory effects, such works intend to increase the user's quality of experience with the application.

The MPEG-V standard provides an architecture and a way to represent information for the communication among the application and the equipment responsible for executing sensory effects. However, it is important to notice that the standard does not define how to specify the application behavior itself. The author is responsible for specifying how to perform the presentation of audiovisual content, its synchronization, and the capture of user interaction.

In order to ease the authoring of mulsemidia applications, this work proposes an extension to the *middleware* Ginga and NCL to provide a declarative approach to the creation of these applications and to allow the authors to reuse the language resources for content synchronization, user interaction, etc. Therefore, this work will extend the definition of media objects in NCL creating a new type of sensory effect's media object. The *middleware* will also be extended in order to define a sensory effect's player which is responsible for controlling the sensory devices during the presentation.

Keywords: Multimedia; Sensory Effects; NCL; MPEG-V

SUMÁRIO

1	Introdução	1
2	Fundamentação Teórica	4
2.1	Padrão MPEG-V	4
2.1.1	Parte I: Arquitetura	4
2.1.2	Parte II: Controle de informação	6
2.1.3	Parte III: Informação sensorial	7
2.1.4	Parte IV: Características de objetos do mundo virtual	7
2.1.5	Parte V: Formato de dados para interações com dispositivos	8
2.1.6	Partes VI e VII	8
2.2	Modelo NCM	9
2.3	Linguagem NCL	10
2.4	Middleware Ginga	11
3	Trabalhos Relacionados	16
4	Desenvolvimento	19
4.1	Efeitos sensoriais em NCL	19
4.2	Reprodutor de efeitos sensoriais	22
4.3	Animação de efeitos sensoriais	24
4.4	Evento de preparação	26
4.5	Caso de uso	27
5	Conclusão	31
	Referências Bibliográficas	32

LISTA DE FIGURAS

FIGURA 1:	Arquitetura simplificada do padrão MPEG-V.	5
FIGURA 2:	Comunicação entre mundo virtual e mundo real [Kim and Han, 2014]	5
FIGURA 3:	Mundo virtual para o mundo virtual [Kim and Han, 2014]	6
FIGURA 4:	Posições dos efeitos sensoriais no padrão MPEG-V.[ISO/IEC, 2016c]	8
FIGURA 5:	Diagrama de classes dos nós de conteúdo do NCM.	9
FIGURA 6:	Diagrama de classes dos nós de âncora do Nested Context Model (NCM).	10
FIGURA 7:	Estrutura do nó raiz da linguagem NCL.	10
FIGURA 8:	Estrutura do cabeçalho da linguagem NCL.	11
FIGURA 9:	Estrutura do corpo da linguagem NCL.	12
FIGURA 10:	Arquitetura do <i>middleware</i> Ginga. Retirado de Lima [2018]	12
FIGURA 11:	Árvore de um documento NCL. Retirado de Lima [2018]	13
FIGURA 12:	Reprodutores de mídia	14
FIGURA 13:	Máquina de estados de evento de NCL	14
FIGURA 14:	Diagrama de classes dos nós de conteúdo do NCM.	17
FIGURA 15:	Posicionamento por coordenadas esféricas.	17
FIGURA 16:	Reprodutores de mídia	22
FIGURA 17:	Diagrama de classes do reprodutor de efeitos sensoriais.	23
FIGURA 18:	Visão geral da aplicação NCL.	27
FIGURA 19:	Esquema de simulação da aplicação com efeitos sensoriais	29
FIGURA 20:	Execução da aplicação com efeitos sensoriais	30

LISTA DE TABELAS

LISTA DE ABREVIACOES

API	Application Programming Interface	16
CIDL	Control Information Description Language	6
DCDV	Device Capability Description Vocabulary	6
DCV	Device Command Vocabulary	8
IIDL	Interaction Interface Description Language	8
IOT	Internet Of Things	16
IPTV	Internet Protocol Television	2, 12
ISDB-T	Sistema Brasileiro De TV Digital Terrestre	2, 12
ITU-T	ITU Telecommunication Standardization Sector	2, 12
MPE	Media Processing Engine	4
MPEG	Moving Picture Experts Group	1
NCL	Nested Context Language	iii, iv, vi, 2, 4, 10, 11, 12, 16, 31, 32
NCM	Nested Context Model	vi, 9, 10, 19
PUC-RIO	Pontifcia Universidade Catlica Do Rio De Janeiro	2, 12
REST	Representational State Transfer	16
SAPV	Sensor Adaption Preference Vocabulary	6
SCDV	Sensor Capability Description Vocabulary	6
SEDL	Sensory Effect Description Language	7, 20, 23, 31
SEM	Sensory Effect Metada	7, 23
SEPV	Sensory Effect Preference Vocabulary	6
SEV	Sensory Effect Vocabulary	7
SIV	Sensed Information Vocabulary	8
SVG	Scalable Vector Graphics	13
UFPB	Universidade Federal Da Paraba	2, 12
UME	Universal Multimedia Experience	1
VE	Virtual Environments	7
XML	Extensible Markup Language	2, 7, 8

Capítulo 1

Introdução

Durante muitos anos, apresentações multimídia se limitaram aos estímulos de apenas dois dos sentidos humanos: a visão e a audição. Esta situação encontra-se em desacordo com o fato de 60% das comunicações humanas serem não-verbais, e que a maioria de nós percebe o mundo pela combinação dos cinco sentidos: visão, audição, tato, paladar e olfato [Ghinea et al., 2014]. Com base nisto, esforços foram angariados para o estudo e a compreensão de como expandir a definição de multimídia para que contemple outros estímulos sensoriais além da visão e da audição.

Recentemente, novas tecnologias como o cinema 4-D surgiram com o propósito de agregar outros estímulos sensoriais às apresentações multimídia. No entanto, o acesso a tais tecnologias ainda é limitado aos cinemas especializados. Esta limitação vai contra a noção de Universal Multimedia Experience (UME), definida em [Pereira and Burnett, 2003]. A definição de UME diz que o usuário deve ter uma experiência informativa equivalente a qualquer hora e em qualquer lugar. O conteúdo deve se adaptar de forma que garanta ao usuário uma experiência significativa. A noção de UME coloca o usuário como foco principal, enquanto fatores como o ambiente e a rede representam apenas veículos para o conteúdo.

Uma forma de tornar as apresentações multimídia com estímulos sensoriais mais acessíveis é por meio da definição de padrões que possibilitem que as informações representadas nestas apresentações sejam interoperáveis entre diversos tipos de sistemas, facilitando a distribuição. A definição de padrões livres teve grande impacto no desenvolvimento de ferramentas que trouxeram o consumo de diferentes tipos de mídia para o cotidiano. Tais padrões possibilitaram que a criação e comunicação de dados digitais destinadas aos sentidos mais importantes – visão e audição – fossem simples, baratas e triviais [Pereira and Burnett, 2003]. Ao se tratar de apresentações multimídia com estímulos sensoriais, também conhecidas com apresentações multemídia, o padrão em vigência é o MPEG-V, definido pelo grupo de padronização internacional Moving Picture Experts Group (MPEG).

O padrão MPEG-V fornece uma arquitetura e especifica a representação das informações associadas para permitir a interoperabilidade entre seus componentes. A arquitetura estabelece

como o fluxo de dados transita entre o mundo virtual, caracterizado pela aplicação multimídia em si, e o mundo real, caracterizado pelo meio onde a aplicação é consumida pelo usuário. A representação das informações é definida por esquemas Extensible Markup Language (XML) que contém dados que especificam características como: as preferências dos efeitos sensoriais, a capacidade dos dispositivos sensores, a descrição dos efeitos sensoriais e as interações com os dispositivos sensores e atuadores.

Apesar de definir os aspectos referentes a representação das informações da aplicação e dos dispositivos sensores e atuadores no ambiente físico, o padrão MPEG-V não especifica a lógica da aplicação, seu controle de sincronização, a captura da interação e a integração com os atuadores, sendo estes de responsabilidade do autor. A fim de facilitar a autoria de aplicações multimídia, este trabalho propõe uma extensão do *middleware* Ginga e da linguagem NCL de forma a prover uma abordagem declarativa para a criação destas aplicações e permitir aos autores reutilizar os recursos da linguagem para a sincronização de conteúdo, interação com o usuário, etc.

O *middleware* Ginga, desenvolvido pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em parceria com a Universidade Federal da Paraíba (UFPB), adotado pelo Sistema Brasileiro de TV Digital Terrestre (ISDB-T) e recomendação ITU Telecommunication Standardization Sector (ITU-T) para serviços Internet Protocol Television (IPTV), implementa um ambiente de apresentação declarativo que possibilita a execução de aplicações multimídia interativas escritas na linguagem NCL. NCL é uma linguagem baseada em XML que define aspectos de interatividade, sincronismo espaço-temporal entre objetos de mídia, adaptabilidade, suporte a múltiplos dispositivos e suporte à produção ao vivo de programas interativos não-lineares.

A extensão da linguagem NCL proposta neste trabalho define novos tipos (atributo *type*) para os objetos de mídia (elementos `<media>`), a fim de permitir a especificação de efeitos sensoriais na aplicação, e novos atributos ao elemento `<region>`, a fim de permitir a especificação das informações de posicionamento dos efeitos sensoriais. Do ponto de vista do *middleware* Ginga, este trabalho propõe a criação de um reprodutor de efeitos sensoriais, que é responsável por controlar os dispositivos sensoriais de acordo com a aplicação. O reprodutor interpreta as informações definidas nos objetos de mídia para gerar uma descrição dos efeitos sensoriais de acordo com o padrão MPEG-V e utiliza esta descrição para enviar mensagens para os dispositivos sensoriais.

O restante desse trabalho está estruturado como segue. O Capítulo 2 apresenta a revisão

da literatura e principais conceitos relacionados ao trabalho sendo desenvolvido. O Capítulo 3 apresenta os trabalhos relacionados. O Capítulo 4 apresenta detalhes da implementação realizada para a extensão do *middleware* Ginga. Por fim, o Capítulo 5 conclui esse trabalho e apresenta trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os principais conceitos relacionados ao desenvolvimento realizado nesse trabalho. Nesse contexto são apresentados o padrão MPEG-V, bem como informações sobre cada uma das suas sete partes. Ainda, é apresentado detalhes da implementação do *middleware* Ginga, bem como informações relativas a linguagem NCL – seus elementos e seu significado.

2.1 Padrão MPEG-V

O padrão MPEG-V fornece uma arquitetura e especifica a representação de informações associadas para possibilitar a interoperabilidade entre o mundo virtual e o mundo real [Kim and Han, 2014]. O mundo virtual é representado pela mídia que é apresentada, podendo ser um filme, jogo, simulação, etc., enquanto o mundo real é representado pelos sensores e atuadores, que transformam a informações digitais em estímulos para o usuário. Este padrão é dividido em 7 partes que serão apresentadas nas seções a seguir. É importante ressaltar, que somente serão apresentados em mais detalhes os vocabulários definidos na partes 3 do padrão, o qual será utilizado na extensão apresentada no Capítulo 4.

2.1.1 Parte I: Arquitetura

Esta parte [ISO/IEC, 2016a] descreve a arquitetura do padrão, os dispositivos de troca de informações e a interoperabilidade entre eles. A Figura 1 ilustra de maneira simples a arquitetura definida pelo padrão MPEG-V. A fonte representa o mundo virtual no qual a mídia está contida, e o usuário representa uma parte do mundo real. O dispositivo de processamento de mídia, denominado Media Processing Engine (MPE), é o dispositivo responsável para realizar a adaptação dos efeitos sensoriais entre o mundo virtual e o mundo real. Os sensores e atuadores são os elementos que realizam a interação entre o mundo real e o mundo virtual com os usuários.

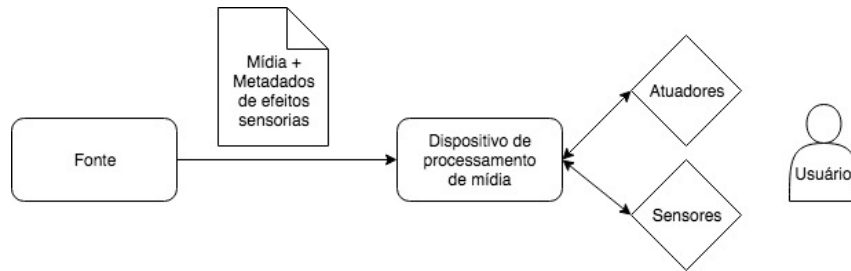


Figura 1: Arquitetura simplificada do padrão MPEG-V.

A Figura 2(a) apresenta a transição de dados do mundo virtual para o mundo real. O dispositivo de adaptação VR (Virtual para Real) recebe os dados de efeitos sensoriais e/ou características de objetos do mundo virtual como entrada, converte estes dados em comandos para o(s) atuador(es), estes comandos, então, são aplicados no mundo real. A Figura 2(b) apresenta a transição de dados do mundo real para o mundo virtual. O dispositivo de adaptação RV (Real para Virtual) converte informações recebidas pelos sensores do mundo real em características de objetos do mundo virtual e/ou informações compreendidas do mundo virtual que são enviados para o mundo virtual.

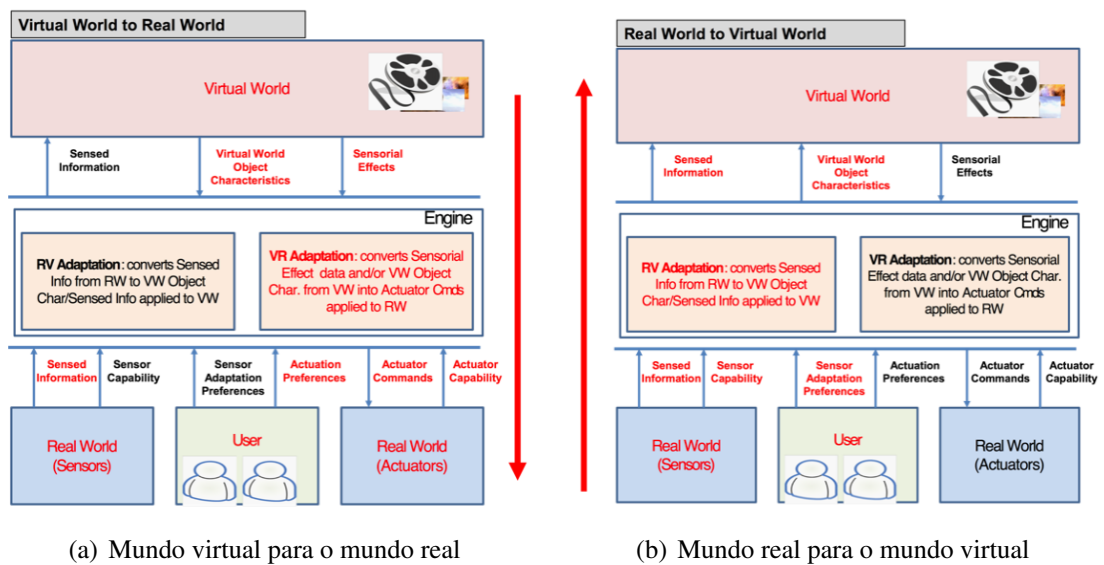


Figura 2: Comunicação entre mundo virtual e mundo real [Kim and Han, 2014]

A Figura 3 apresenta a transição de dados dentro do mundo virtual. Essa é a parte em que as informações provenientes do mundo real interagem com as informações presentes no mundo virtual.

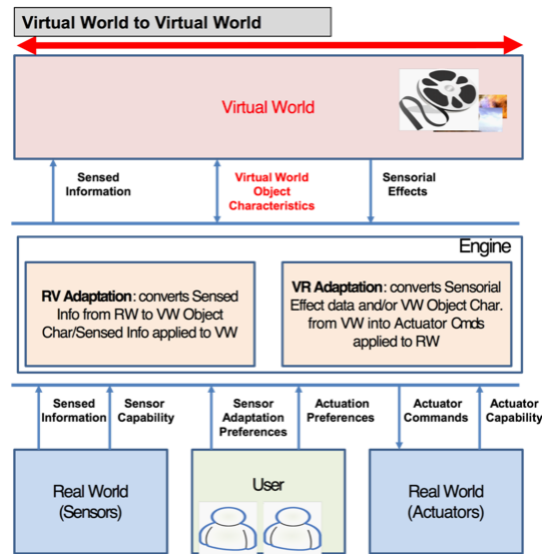


Figura 3: Mundo virtual para o mundo virtual [Kim and Han, 2014]

2.1.2 Parte II: Controle de informação

Esta parte [ISO/IEC, 2016b] especifica a sintaxe e a semântica das ferramentas requeridas para providenciar a interoperabilidade nos dispositivos de controle (atuadores e sensores) no mundo real e no mundo virtual [Kim and Han, 2014]. Nesta parte é especificada uma linguagem de controle de informação, chamada Control Information Description Language (CIDL), em formato XML. Na definição da linguagem são especificados vocabulários referentes a descrição da capacidade dos dispositivos por meio do Device Capability Description Vocabulary (DCDV), descrição de capacidades dos sensores por meio do Sensor Capability Description Vocabulary (SCDV), preferências de efeitos sensoriais por meio do Sensory Effect Preference Vocabulary (SEPV), e preferências dos sensores por meio do Sensor Adaption Preference Vocabulary (SAPV).

O DCDV descreve as capacidades dos atuadores como, por exemplo, intensidade máxima de uma lâmpada ou de um efeito de vibração. O SCDV descreve as capacidades dos sensores como, por exemplo, máxima temperatura de operação de termômetros. O SEPV especifica as preferências do usuário sobre os efeitos sensoriais como, por exemplo, a máxima intensidade permitida para uma lâmpada. O SAPV especifica as preferências do usuário sobre as informações sensoriais como, por exemplo, um sensor de som só pode captar uma certa variação de frequência. A CIDL tem o objetivo de fornecer informações sobre os dispositivos sensoriais do mundo real e preferências do usuário para o dispositivo de adaptação, de forma que auxilie a

interação com os dispositivos sensoriais disponíveis e se ajuste de acordo com as preferencias do usuário.

2.1.3 Parte III: Informação sensorial

Nesta parte [ISO/IEC, 2016c] é especificada a linguagem Sensory Effect Description Language (SEDL) em formato XML que possibilita descrever efeitos sensoriais que estimulam os sentidos humanos [Kim and Han, 2014]. Nesta linguagem é especificado um vocabulário em que são definidos os tipos de efeitos sensoriais, chamado Sensory Effect Vocabulary (SEV), além de permitir que cada domínio de aplicação defina seus próprios efeitos sensoriais. Uma descrição de acordo com a SEDL é denominada Sensory Effect Metada (SEM) e é agregada ao conteúdo multimídia apresentado para controlar os atuadores proporcionando estímulos sensoriais apropriados ao usuário.

A Listagem 2.1 ilustra um exemplo da SEDL, no exemplo é definido um efeito sensorial de luz com intensidade de 50 lux, dentre uma variação de 0.0001 a 32000 lux, na posição *left:middle:front*, com a cor *#FF0000*, que é ativado em *pts="0"* e desativado em *pts="28"*. O exemplo da Listagem 2.1 indica que o atuador de luz se encontra na posição frontal, no meio a esquerda do usuário.

```

1 <Effect type="LightType" intensity-value="50.0" intensity-range="0.00001 32000.0" activate="
  true" location="left:middle:front" color="#FF0000" pts="0"/>
2 <Effect type="LightType" activate="false" location="left:middle:front" pts="28"/>

```

Listing 2.1: Exemplo da SEDL [ISO/IEC, 2016c].

No padrão MPEG-V, o posicionamento dos efeitos sensoriais é definido por pontos nos eixos x, y e z, conforme ilustra a Figura 4. O eixo x pode ter os valores: *left*, *centerleft*, *center*, *centerright* ou *right*. O eixo y pode ter os valores: *top*, *middle* ou *bottom*. O eixo z pode ter os valores: *front*, *midway* ou *back*. Além dos valores padrão, os eixos podem ter o valor *, que significa que o efeito sensorial ocorre em todos os pontos do eixo.

2.1.4 Parte IV: Características de objetos do mundo virtual

Nesta parte [ISO/IEC, 2016d] são definidos tipos de atributos e características de objetos do mundo virtual. Uma especificidade dos ambientes virtuais, do inglês Virtual Environments (VE), em relação a outras aplicações multimídia consiste na representação de objetos do mundo

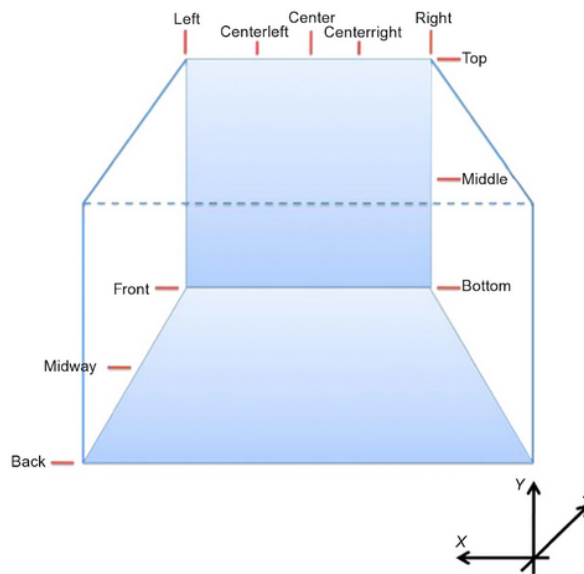


Figura 4: Posições dos efeitos sensoriais no padrão MPEG-V.[ISO/IEC, 2016c]

virtual dentro do ambiente. Um objeto do mundo virtual pode ser classificado em dois tipos: avatares e objetos genéricos [Kim and Han, 2014]. A motivação da representação de objetos do mundo virtual é prover ao usuário a sensação de presença no mundo virtual e fornecer meios de interação com o mundo virtual.

2.1.5 Parte V: Formato de dados para interações com dispositivos

Nesta parte [ISO/IEC, 2016e] é especificada a sintaxe e a semântica dos formatos de dados para a interação entre os dispositivos (sensores e atuadores) requeridos para providenciar a interoperabilidade no controle dos dispositivos e nas informações sensoriais dos dispositivos de interação no mundo real e virtual [Kim and Han, 2014]. Nesta parte é especificada uma linguagem de interação de interfaces, chamada Interaction Interface Description Language (IIDL), em formato XML. Na definição da linguagem são especificados vocabulários para o controle de dispositivos, chamado Device Command Vocabulary (DCV), e informação compreendida, chamado Sensed Information Vocabulary (SIV). O DCV descreve o formato das informações para controlar os atuadores. O SIV define o formato das informações captadas por um sensor.

2.1.6 Partes VI e VII

Na parte VI [ISO/IEC, 2016f] é especificada a sintaxe e semântica dos tipos de dados e as ferramentas definidas nas outras parte [Kim and Han, 2014]. Os tipos de dados especificados

nesta parte servem como base para as ferramentas descritas anteriormente, sendo apresentados de uma maneira geral para que possam compreender os diferentes casos de uso. Na parte VII [ISO/IEC, 2016g] estabelece-se formas de validação e esclarecimento das especificações das partes descritas anteriormente, e testes de conformidade para checar se as aplicações são compatíveis com o padrão MPEG-V.

2.2 Modelo NCM

A definição de documentos hipermídia no NCM é baseada nos conceitos de nós e elos. Nós (*nodes*) representam a informação veiculada em uma aplicação e elos (*links*) são usados para a definição de relacionamentos entre os nós. Um nó é uma entidade NCM que tem como propriedades básicas: um conteúdo, um descritor genérico e uma lista ordenada de âncoras. O modelo distingue duas classes básicas de nós, chamados de nós de conteúdo (*content nodes*) e nós de composição (*composite nodes*). Os nós de conteúdo (*content node*), também chamados nós de mídia ou objetos de mídia, são nós que representam uma mídia qualquer [Soares and Rodrigues, 2005].

Nós de conteúdo são especializados em subclasses que definem o tipo do conteúdo da mídia para que a mídia possa ser interpretada. As subclasses dos nós de conteúdo podem ser: nós de texto, nós de imagem, nós de áudio, nós de vídeos, nós de aplicação, nós de tempo e nós de configuração [Soares and Rodrigues, 2005]. A Figura 14 ilustra o diagrama de classes dos nós de conteúdo com as subclasses previamente definidas.

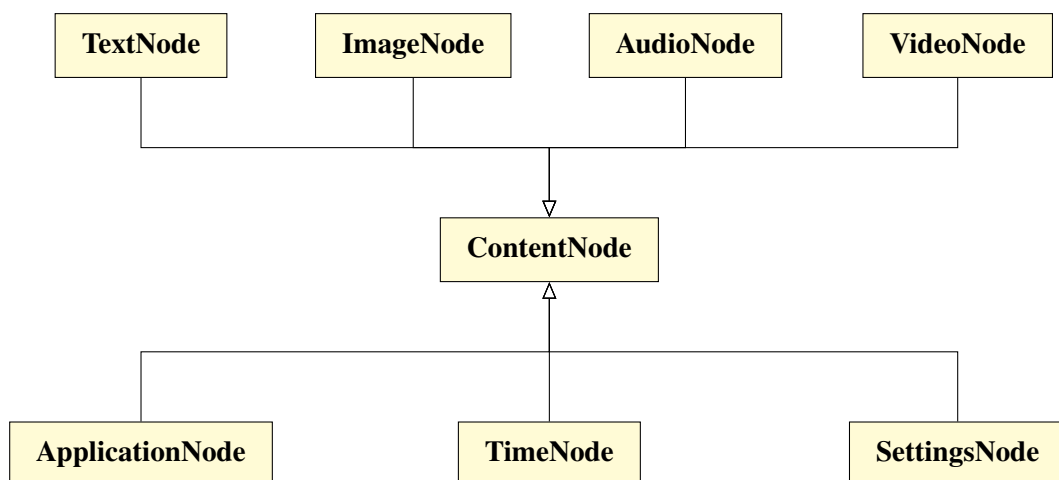


Figura 5: Diagrama de classes dos nós de conteúdo do NCM.

Uma âncora (*anchor*) é uma entidade NCM que pode ser especializada como âncora de

conteúdo (*content anchor* ou apenas âncora) ou âncora de atributo (*attribute anchor* ou propriedade) [Soares and Rodrigues, 2005], conforme ilustra o diagrama de classes da Figura 6.

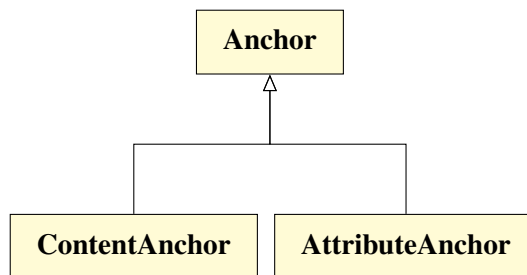


Figura 6: Diagrama de classes dos nós de âncora do NCM.

Os descritores agrupam informações das características de apresentação de objetos de mídia com o propósito de separar as informações do conteúdo da mídia e de sua forma de apresentação [Soares and Rodrigues, 2005]. As informações contidas nos descritores são utilizadas para inicializar as propriedades de uma mídia. As propriedades define conjuntos de chave e valor que representam as informações de uma mídia.

2.3 Linguagem NCL

A linguagem NCL, é uma linguagem declarativa, baseada no modelo conceitual NCM, utilizada para o desenvolvimento de aplicações multimídia interativas com sincronização espaço-temporal de objetos de mídia. É a linguagem adotada pelo *middleware* Ginga para o desenvolvimento de aplicações para o sistema brasileiro de TV digital.

A estrutura básica da linguagem consiste de um nó raiz, determinado pelo elemento `<ncl>`, que possui um cabeçalho, determinado pelo elemento `<head>`, e um corpo, determinado pelo elemento `<body>` [Soares et al., 2009], conforme ilustra a Figura 7. No cabeçalho são definidas as regiões, que especificam a posição onde uma mídia será apresentada, os descritores, que definem como as mídias serão apresentadas, e os conectores, que definem relações genéricas. No corpo são definidos os contextos, objetos de mídia, elos que instanciam os conectores, dentre outros elementos.

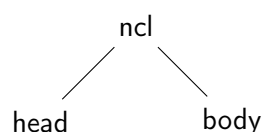


Figura 7: Estrutura do nó raiz da linguagem NCL.

Os principais componentes do cabeçalho de um documento NCL são os elementos `<regionBase>`, `<descriptorBase>` e `<connectorBase>` [Soares et al., 2009], conforme ilustra a Figura 8. O elemento `<regionBase>` define um grupo de regiões através dos elementos `<region>`, que especificam as regiões da tela onde uma mídia poderá ser apresentada. O elemento `<descriptorBase>` define um grupo de descritores através dos elementos `<descriptor>`, que especificam os valores iniciais das propriedades dos objetos de mídia. O elemento `<connectorBase>` define um conjunto de relações através de conectores especificados pelo elementos `<causalConnector>`. Os elementos componentes do cabeçalho podem possuir um elemento `<importBase>` que é utilizado para importar definições contidas em um documento externo, permitindo o reuso destas definições.

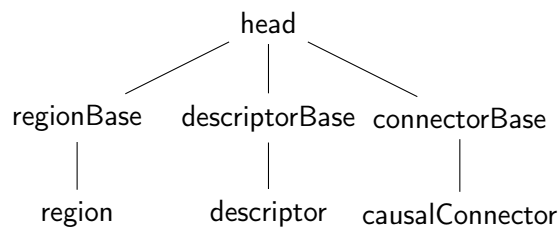


Figura 8: Estrutura do cabeçalho da linguagem NCL.

Os principais componentes do corpo de um documento NCL são os elementos `<port>`, `<context>`, `<media>`, `<link>` e `<switch>` [Soares et al., 2009], conforme ilustra a Figura 9. O elemento `<port>` é a entrada do documento NCL, e especifica um mapeamento para um nó do documento pelo qual a execução se iniciará. O elemento `<context>` permite a estruturação de um documento NCL, sendo o corpo do documento é um tipo especial de contexto. Portanto, nós de contexto possuem os mesmo elementos e características que o corpo de um documento NCL. O elemento `<media>` representa um objeto de mídia, contendo a referência para um descritor e âncoras de conteúdo e de propriedades. O elemento `<link>` define as relações entre objetos de mídia e de composição, contendo a referência para um conector e a vinculação dos objetos. O elemento `<switch>` permite a definição de alternativas entre objetos `<media>`, `<context>` ou outros elementos `<switch>`, que são avaliadas durante a apresentação.

2.4 Middleware Ginga

Um *middleware* é um software que se encontra entre o sistema operacional e as aplicações que são executadas nele. Ele funciona como uma camada de abstração que gerencia os recursos de baixo nível do sistema operacional, como as redes de comunicação, acesso a periféricos do

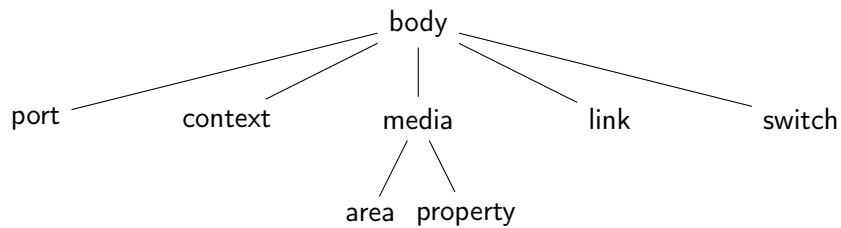


Figura 9: Estrutura do corpo da linguagem NCL.

sistema, renderização de gráficos, dentre outros. Essa camada permite que as aplicações que são executadas no *middleware* possuam uma interface de mais alto nível. Ginga é um *middleware* utilizado para o desenvolvimento de aplicações para TV digital.

O *middleware* Ginga, desenvolvido pela PUC-Rio em parceria com a UFPB, adotado pelo ISDB-T e recomendação do ITU-T para serviços IPTV, implementa um ambiente de apresentação declarativo que possibilita a execução de aplicações multimídia interativas escritas na linguagem NCL dando suporte a múltiplos dispositivos e suporte à produção ao vivo de programas interativos não-lineares.

A Figura 10 ilustra a arquitetura da implementação de referência¹ do *middleware* Ginga, composta por uma API externa e uma API interna. A API externa define a classe abstrata Ginga, a qual é utilizada para expor as funcionalidades do *middleware* e auxiliar na implementação de interfaces gráficas. A API interna implementa as funcionalidades do *middleware*, e é composta pelas classes *Formatter* (formatador), *Parser* (analisador sintático e semântico) e *Document* (documento).

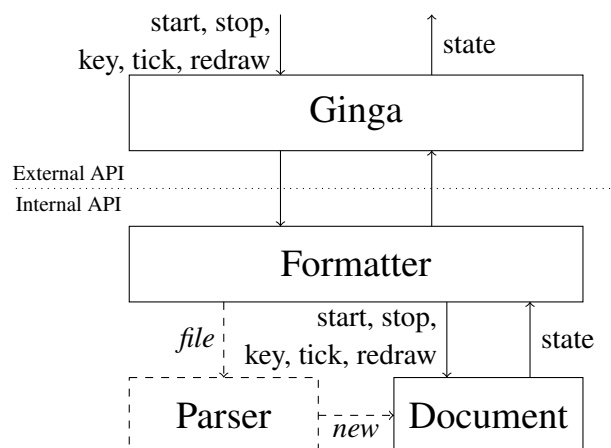


Figura 10: Arquitetura do *middleware* Ginga. Retirado de Lima [2018]

O formatador controla o ciclo de vida de uma apresentação, propagando ações decorrentes

¹Utilizou-se como base a implementação do *middleware* Ginga disponível em <https://github.com/TeleMidia/ginga>

de eventos da API externa, como interação do usuário, alterações no estado de execução dos players de mídia e o avanço no tempo da apresentação, para o documento. Ao iniciar a execução do *middleware* para reprodução de uma aplicação NCL, o formatador instancia o *parser* para a análise do documento NCL fornecido e a criação de um objeto do tipo Document, que representa a árvore de objetos derivada da especificação da aplicação. O objeto Document atua como uma interface de alto nível para a árvore do documento NCL.

A árvore do documento é composta por objetos que são especializados em nós de mídia e nós de composição, conforme ilustrado na Figura 11. Os objetos do tipo Media representam os elementos <media> de um documento NCL, e podem ser especializados em um objeto do tipo MediaSettings, representando elementos <media> do tipo *settings*. Os nós de composição são especializados nos objetos do tipo Context, que representam os elementos <context>, e objetos do tipo Switch, que representam os elementos <switch> de um documento NCL.

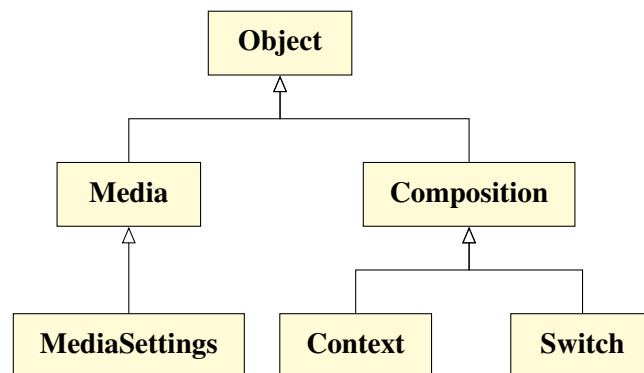


Figura 11: Árvore de um documento NCL. Retirado de Lima [2018]

Os objetos de mídia do documento são responsáveis pela reprodução de diferentes tipos de mídia, por exemplo: imagens, *scripts* Lua, Scalable Vector Graphics (SVG), texto ou vídeo. Para isto, cada objeto de mídia mantém um reproduzidor (classe Player) que é especializado no tipo de mídia referente ao objeto, conforme ilustra o diagrama da Figura 12. Os *players* são responsáveis por implementar a lógica necessária para a execução do tipo de mídia designado, além de controlar a apresentação da mídia.

No início da execução de um documento NCL, o formatador inicializa (ação *start*) o nó raiz da árvore do documento, correspondendo ao corpo do documento NCL. Esta ação é então propagada para os outros nós da árvore, disparando outras ações e modificando o estado do documento. Após a inicialização do documento, o formatador, controlado pela API externa, pode propagar outras ações para serem avaliadas sobre o documento. Estas ações são originadas a partir da ocorrência de eventos como: seleção de tecla (*key*), avanço no tempo de apresentação

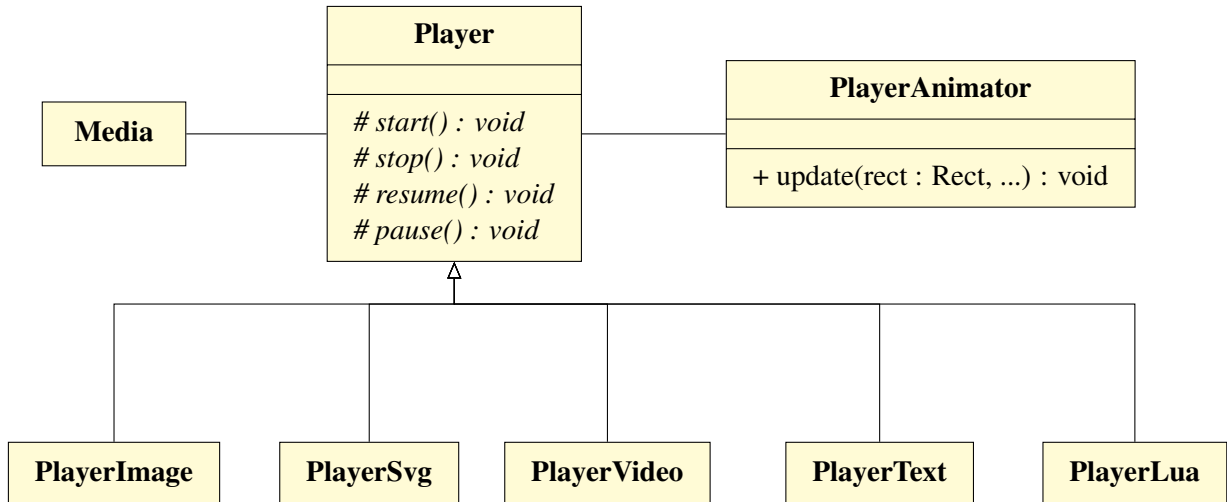


Figura 12: Reprodutores de mídia

da aplicação (*tick*), solicitação de uma nova renderização de *frames* (*redraw*) ou a finalização da apresentação da aplicação (*stop*).

Em uma aplicação NCL, cada objeto do documento está associado a um conjunto de eventos, e cada evento contém um estado representado pela máquina de estados de eventos de NCL, conforme ilustrado na Figura 13. Um evento pode ser do tipo *Presentation*, representando ocorrências relacionadas a apresentação de um objeto de mídia; *Attribution*, representando a modificação do valor de uma propriedade de um objeto de mídia; ou do tipo *Selection*, representando a interação do usuário com um objeto usando o mouse ou alguma tecla.

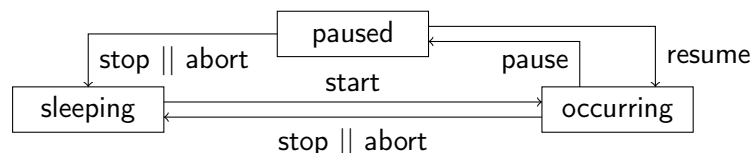


Figura 13: Máquina de estados de evento de NCL

Quando uma ação é propagada sobre a árvore do documento, os objetos reagem a esta ação efetuando a transição de estado dos eventos designados pela ação. Caso a transição de estado de um evento seja bem sucedida e existam elos (*links*) relacionados à ação, estes são acionados, desencadeando a execução de novas ações sobre o documento.

As ações decorrentes de eventos de apresentação são propagadas até os reprodutores que controlam a apresentação através de métodos homônimos às ações que inicializam (*start*), pausam (*pause*), retomam (*resume*) ou encerram (*stop*) a mídia alvo da ação. Nos casos de eventos de atribuição, cada reprodutor mantém um “animador”, definido pela classe *PlayerAnimator* (ver Figura 12), que é responsável por controlar a atualização de suas propriedades. A cada

nova renderização do objeto de mídia, *e.g.*, de um dos seus *frames*, seu reprodutor invoca o método *update* no animador atualizando as propriedades da mídia.

Capítulo 3

Trabalhos Relacionados

Este capítulo apresenta trabalhos relacionados à extensão da linguagem NCL ou do *middleware* Ginga para a inclusão de efeitos sensoriais em aplicações multimídia e ferramentas para a reprodução de aplicações multimedial.

Saleme and Santos [2015] apresentam a plataforma *PlaySEM*, uma abordagem de reprodutor de efeitos sensoriais compatível com MPEG-V que opera independente de um reprodutor de mídia audiovisual. A plataforma, desenvolvida em Java, é composta por 3 componentes: *SE Video Player*, responsável pela reprodução de vídeo e entrada dos metadados de efeitos sensoriais; *SE Renderer*, responsável pela transformação de metadados de efeitos sensoriais em comandos e acionamento dos dispositivos que renderizarão os efeitos sensoriais, e *Device Control*, vinculado ao *SERenderer*, um programa embarcado em um microcontrolador que recebe os comandos para acionar os dispositivos físicos. Os 3 softwares são utilizados em conjunto para reproduzir vídeos com anotações de efeitos sensoriais e renderizá-los no ambiente do usuário. O usuário fornece como entrada um conteúdo de mídia e os metadados de efeitos sensoriais no padrão MPEG-V para reprodução no *SE Video Player*, que se comunica com o *SE Renderer* para o processamento dos metadados, que por sua vez, envia comandos para o *Device Control*, que controla os dispositivos sensoriais.

Guedes et al. [2016] investigam o uso de uma linguagem multimídia, no caso NCL, para especificar as relações entre objetos de mídia, sensores e atuadores. Na abordagem utilizada pelos autores são definidos objetos de mídia de *scripts* Lua que acessam via HTTP com uma Application Programming Interface (API) Representational State Transfer (REST) mantida por dispositivos Internet of Things (IoT). Através desta API REST são controlados os dispositivos sensoriais.

Em Josué et al. [2018a], é proposta a modelagem de efeitos sensoriais como entidades de primeira classe representadas por nós. Para possibilitar a modelagem de efeitos sensoriais é proposta a adição de uma subclasse aos nós de conteúdo de NCM. Esta modificação permite especializar os nós de conteúdo em objetos de mídia que representam efeitos sensoriais. Tais nós são denominados de nó de efeitos sensoriais. A Figura 14 ilustra o diagrama de classes dos

nós de conteúdo com as subclasses previamente definidas e a subclasse de efeitos sensoriais, destacada em laranja.

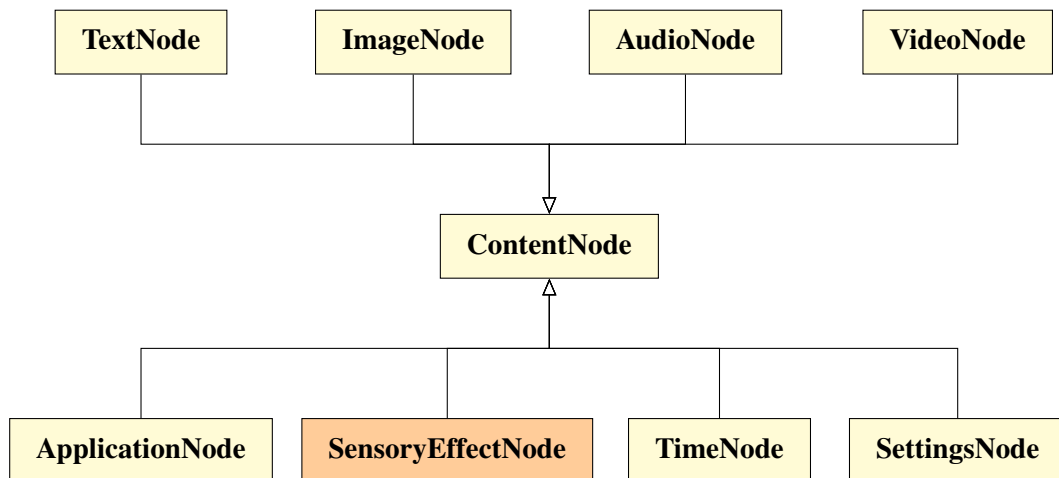
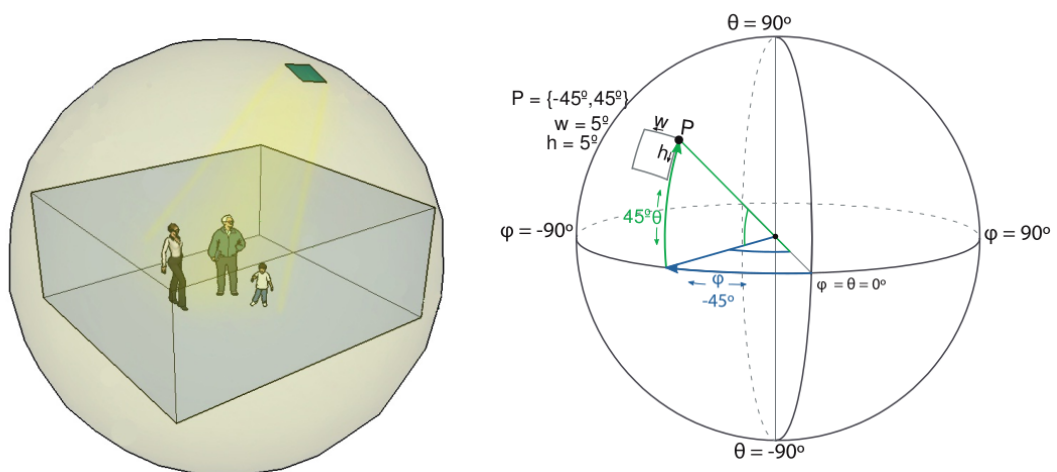


Figura 14: Diagrama de classes dos nós de conteúdo do NCM.

Além da representação de efeitos sensoriais como nós, em [Josué et al., 2018a] foi proposta uma outra abordagem, além da definida pelo padrão MPEG-V, para definir o posicionamento de efeitos sensoriais. Esta proposta considera que o ambiente físico esteja circunscrito por uma esfera, conforme representado na Figura 15(a). Desta forma, são utilizadas as coordenadas esféricas para definir o posicionamento dos efeitos sensoriais.



(a) Ambiente físico circunscrito em uma esfera [Josué et al., 2018a].

(b) Sistema de coordenadas esféricas [Josué et al., 2018a].

Figura 15: Posicionamento por coordenadas esféricas.

O sistema de coordenadas esféricas usa dois atributos para definir a posição de um ponto: o ângulo polar (ϕ), medido em relação ao eixo z, e o ângulo azimutal (θ), medido em relação ao

eixo x, conforme representado na Figura 15(b). Além dos atributos polar e azimutal, também são definidos os atributos *width*, indicando uma largura, e *height*, indicando uma altura, ambos em graus, para especificar o tamanho da região do efeito sensorial.

A vantagem de utilizar o sistema de coordenadas esféricas para definir o posicionamento de efeitos sensoriais é a possibilidade de definir a localização mais precisamente. O esquema de posicionamento definido pelo padrão MPEG-V especifica apenas alguns pontos pré-definidos para a localização de efeitos sensoriais, enquanto o sistema de coordenadas esféricas permite definir diversos pontos, além de, por definição, especificar a direção do efeito sensorial em ângulos e possibilitar determinar pequenas variações na posição dos efeitos sensoriais. Ao permitir a definição de pequenas variações na posição dos efeitos sensoriais, o sistema de coordenadas esféricas facilita a criação de animações utilizando as posições dos efeitos sensoriais, que seriam mais difíceis utilizando o formato de posicionamento do padrão MPEG-V, onde os pontos pré-definidos são mais dispersos.

Capítulo 4

Desenvolvimento

4.1 Efeitos sensoriais em NCL

Parte do desenvolvimento realizado nesse trabalho envolve permitir a definição de efeitos sensoriais em aplicações NCL. A partir da noção de nós do NCM é possível modelar efeitos sensoriais como nós de conteúdo e relacioná-los com outros tipos de nós por meio de elos, como feito em [Josué et al., 2018a]. Dessa forma, utiliza-se de toda a capacidade de sincronização das mídias definida pelo NCM para a definição da aplicação. A vantagem do NCM para a modelagem de aplicações multimídia com múltiplos efeitos é a compatibilidade da estrutura do modelo com o padrão MPEG-V. É possível utilizar os próprios recursos do NCM para definir informações sobre os efeitos sensoriais que serão utilizadas para gerar os metadados de efeitos sensoriais de acordo com o padrão MPEG-V.

A definição de nós de efeitos sensoriais é feita como os demais nós que representam objetos de mídia. Nós de efeitos sensoriais são definidos explicitamente ao declarar o tipo do objeto de mídia e são utilizadas as informações das propriedades e/ou dos descritores desse nó para identificar as informações dos efeitos sensoriais.

Do ponto de vista da linguagem NCL, os objetos de mídia de efeitos sensoriais possuem o atributo *type*, representado pela string "*application/x-sensory-effect*", concatenado com o tipo do efeito sensorial e separado por hífen. No exemplo da Listagem 4.1, a string "*application/x-sensory-effect-WindType*" representa um objeto de mídia de efeito sensorial do tipo vento.

```
1 <media id="media_2" type="application/x-sensory-effect-WindType" />
```

Listing 4.1: Objeto de mídia de efeito sensorial do tipo vento.

De acordo com a modelagem de nós de efeitos sensoriais no NCM, o objeto de mídia pode definir as informações sobre o efeito sensorial por meio de propriedades. As propriedades de objetos de mídia representando efeitos sensoriais correspondem as propriedades dos efeitos sensoriais. Estas propriedades são análogas as propriedades definidas pelo padrão MPEG-V [ISO/IEC, 2016c].

Assim, as propriedades devem definir valores correspondentes as propriedades de efeitos sensoriais especificadas pela SEDL, definida na parte 3 do padrão MPEG-V [ISO/IEC, 2016c], como apresentado na Seção 2.1.3.

```

1 <media id="media_3" type="application/x-sensory-effect-WindType">
2   <property name="location" value="left:top:*"/>
3   <property name="intensity" value="0.3"/>
4 </media>

```

Listing 4.2: Objeto de mídia de efeito sensorial do tipo vento com propriedade.

A Listagem 4.2 mostra um exemplo de objeto de mídia de efeito sensorial do tipo vento com as propriedades *location*, que indica a posição do efeito sensorial no formato definido pelo padrão MPEG-V, e *intensity*, que indica a intensidade do efeito sensorial.

Além das propriedades, é possível utilizar os descritores para definir os atributos de objetos de mídia de efeitos sensoriais. Uma vantagem do uso de descritores é a possibilidade de reutilizar o mesmo descritor para diferentes objetos de mídia. Em apresentações multimídia com efeitos sensoriais pode ser que hajam, por exemplo, atuadores que estejam na mesma posição e operam com a mesma intensidade, ou seja, compartilham de informações comuns. A Listagem 4.3 ilustra um caso em que os efeitos sensoriais dos tipos vento e luz compartilham as mesmas informações de posição e intensidade. Então, é definido um descritor único ao qual os objetos de mídia são associados. O descritor especificado na figura define um parâmetro com a intensidade e a posição através de uma região, que representa a localização *"left:top:front"* no formato definido pelo padrão MPEG-V.

```

1 <ncl>
2   <head>
3     <regionBase>
4       <region id="reg_0" x-axis="left" y-axis="top" z-axis="front"/>
5       <!-- ... -->
6     </regionBase>
7     <descriptorBase>
8       <descriptor id="desc_1" region="reg_0">
9         <descriptorParam name="intensity" value="0.3"/>
10      </descriptor>
11     <!-- ... -->
12   </descriptorBase>
13   <!-- ... -->
14 </head>
15 <body>
16   <!-- ... -->
17   <media id="media_4" type="application/x-sensory-effect-WindType" descriptor="desc_1"/>

```

```

18     <media id="media_5" type="application/x-sensory-effect_LightType" descriptor="desc_1"
19     />
19     <!-- ... -->
20 </body>
21 </ncl>

```

Listing 4.3: Diferentes objetos de mídia de efeito sensorial associados a um mesmo descritor.

Os atributos de posicionamento do elemento `<region>` definem uma área da tela por meio dos atributos `left`, `top`, `width` e `height`. Apesar destes atributos serem suficientes para definir a posição de objetos de mídia bidimensionais como vídeos, imagens ou texto, estes não se aplicam para o posicionamento de efeitos sensoriais. Para definir o posicionamento de efeitos sensoriais é proposto estender a lista atributos do elemento `<region>` para que contemple os formatos de posicionamento definidos para representar os efeitos sensoriais.

Para definir o posicionamento dos efeitos sensoriais em um elemento `<region>` de acordo com o formato estabelecido pelo padrão MPEG-V, foram incluídos os atributos `x-axis`, `y-axis` e `z-axis` (ver a Listagem 4.3). Caso o valor do eixo não seja definido será atribuído o valor `*` a esse eixo. A especificação dos eixos em uma região é utilizada para definir a propriedade `location` de um objeto de mídia de efeitos sensoriais. A propriedade `location` é definida pela concatenação dos valores dos atributos `x-axis`, `y-axis` e `z-axis`, separados por dois-pontos, conforme é especificado pelo padrão MPEG-V. A propriedade `location` definida a partir dos valores dos eixos definidos na região da Listagem 4.3 será `"left:top:front"`.

Para definir o posicionamento de efeitos sensoriais pelo sistema de coordenadas polares em um elemento `<region>`, foram incluídos os atributos `polar`, que indica o ângulo polar, e `azimuthal`, que define o ângulo azimutal, conforme mostra a Listagem 4.4. Os valores dos atributos devem possuir a extensão `deg` para indicar que os valores são definidos em graus. Os atributos `width` e `height` são definidos por padrão na NCL, no entanto estes valores são definidos por padrão em porcentagem ou `pixels`. Para definir a região de um efeitos sensorial estes valores também devem possuir a extensão `deg` para indicar que os valores estão em graus.

```

1 <region id="reg_1" polar="45deg" azimuthal="45deg" width="5deg" height="5deg"/>

```

Listing 4.4: Região definida com coordenadas esféricas.

Assim como na região definida no formato do padrão MPEG-V, os atributos da região no sistema de coordenadas esféricas serão utilizados para construir a propriedade `location` dos objetos de mídia que estiverem associados a região. A propriedade `location` para regiões definidas pelo sistema de coordenadas esféricas será estabelecida pela concatenação dos valores dos atributos

polar, *azimuthal*, *width* e *height*, sem a extensão *deg* e separados por vírgula. A propriedade *location* do exemplo descrito na Figura 4.4 será “45,45,5,5”. Os atributos *polar* e *azimuthal* são obrigatórios e os atributos *width* e *height* são opcionais.

4.2 Reprodutor de efeitos sensoriais

A implementação dos *players* de conteúdo de mídia no *middleware* Ginga possuem uma arquitetura semelhante ao padrão de projeto *Bridge* [Gamma et al., 1995]. Os objetos de mídia da árvore do documento mantêm um reprodutor definido pela classe *Player*, o qual é especializado para cada tipo de objeto de mídia conforme descrito na Seção 2.4. Este aspecto da arquitetura do *middleware* Ginga possibilita que a abstração do *player* seja desacoplada da implementação da lógica para a reprodução de conteúdos de mídia, facilitando a integração de novos *players*.

Dadas as características arquiteturais do *middleware* Ginga, este trabalho desenvolveu a classe *PlayerSE* (destacada em laranja na Figura 16). Esta classe é uma especialização da classe *Player*, e implementa as funcionalidades necessárias para possibilitar a renderização de efeitos sensoriais. O *player* de efeitos sensoriais deve ser capaz de lidar com dispositivos heterogêneos, que podem executar diferentes sistemas operacionais e estarem conectados a uma variedade de protocolos de comunicação [Saleme et al., 2018].

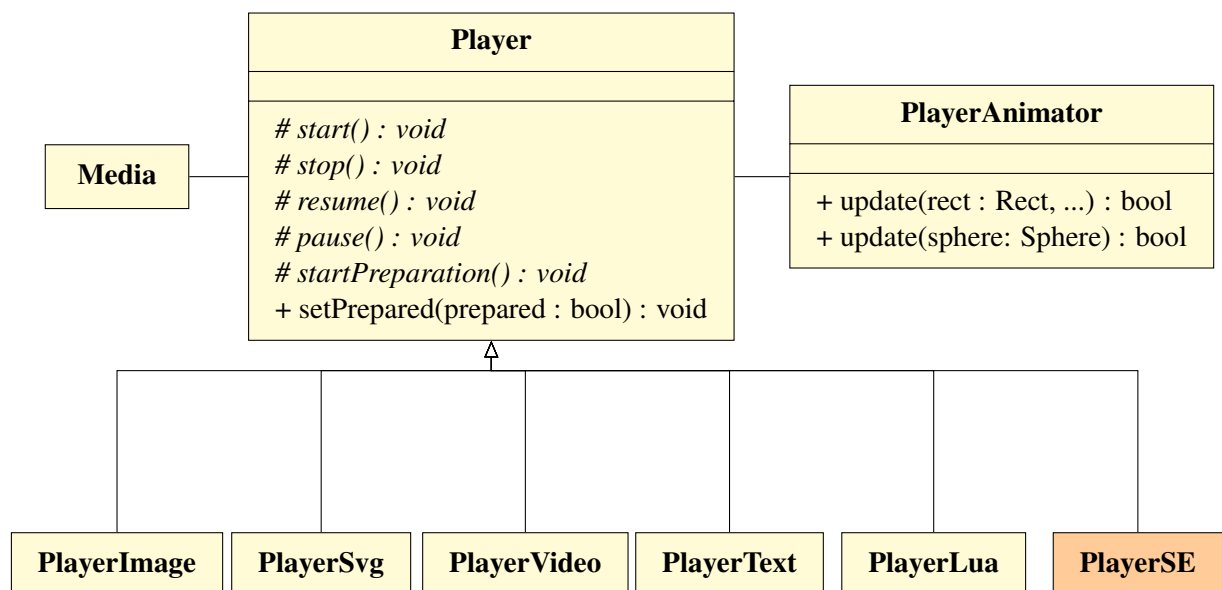


Figura 16: Reprodutores de mídia

O *player* de efeitos sensoriais aborda a questão da integração com dispositivos heterogêneos

utilizando um modelo de dados compatível com o padrão MPEG-V. A utilização de um padrão livre, possibilita a representação das informações na aplicação de maneira interoperável entre diversos tipos de sistemas. Desse modo, o *player* de efeitos sensoriais, proposto neste trabalho, processa as informações contidas nas propriedades do objeto de mídia e aquelas relativas ao evento realizado sobre o objeto de mídia, a fim de obter os metadados dos efeitos sensoriais e gerar uma descrição SEM de acordo com a linguagem SEDL.

Para gerenciar a conexão com diversos protocolos de comunicação, o *player* de efeitos possui uma estrutura semelhante ao padrão *Adapter* [Gamma et al., 1995]. Cada *player* de efeito sensorial mantém um protocolo, definido pela interface `Protocol`, que especifica alguns métodos básicos de protocolos de comunicação, conforme apresentado no diagrama de classes da Figura 17. Desta forma, é possível estabelecer uma estrutura comum para a criação de classes que encapsulam a interface de bibliotecas que implementam os protocolos de comunicação, e torná-las acessíveis ao *player*, facilitando a integração de novos protocolos.

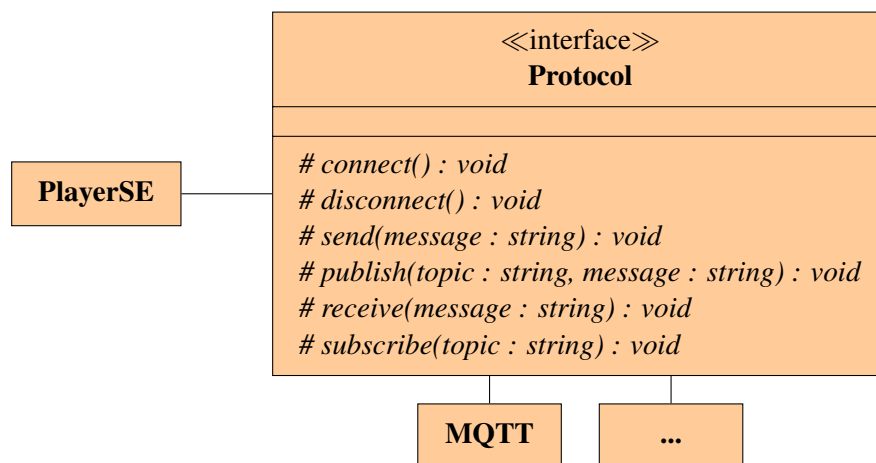


Figura 17: Diagrama de classes do reprodutor de efeitos sensoriais.

A fim de demonstrar a utilização da solução proposta, este trabalho utiliza o protocolo MQTT [Banks and Gupta, 2014], um protocolo de mensagens leve baseado no modelo *publisher-subscriber*, para a comunicação com os dispositivos atuadores. Para isto, foi criada a classe MQTT (Figura 17), que encapsula as funcionalidades da biblioteca Eclipse Paho, uma implementação *open-source* do protocolo em C++. A integração com o protocolo MQTT funciona a partir da publicação (método *publish*) de mensagens contendo a descrição SEM no tópico definido a partir do valor da propriedade *location*, conforme especificada na Seção 4.1, dos objetos de mídia quando ocorrem eventos NCL. Desta forma, um dispositivo atuador deve se inscrever (método *subscribe*) no tópico referente a seu posicionamento para receber as mensagens do

reprodutor com a descrição SEM e renderizar os efeitos sensoriais.

4.3 Animação de efeitos sensoriais

Em NCL, a modificação de uma propriedade de um objeto de mídia é feita por meio de eventos de atribuição disparados a partir de elos especificados no documento NCL. Um evento de atribuição em NCL pode ser instantâneo ou realizado ao longo de um intervalo de tempo, onde o tempo é incrementado de acordo com um valor específico definido no próprio evento de atribuição. Utilizando este conceito, o autor de uma aplicação é capaz de especificar uma animação em NCL, definindo eventos de atribuição sobre as propriedades de posicionamento ou tamanho ao longo de um intervalo de tempo. Tais animações são definidas usando os atributos *duration*, para a duração, e *by*, para o incremento, de um conector NCL.

Para utilizar as estruturas definidas pela linguagem NCL para a especificação de animações de efeitos sensoriais foram necessárias algumas modificações na implementação de referência¹ do *middleware* Ginga a fim de dar suporte às propriedades de posicionamento de efeitos sensoriais e ao atributo *by*, que não estava implementado. Como foi dito anteriormente, no presente trabalho, o posicionamento de um objeto de mídia representando um efeito sensorial pode ser definido através dos eixos *x*, *y* e *z*, conforme definido pelo padrão MPEG-V; ou por coordenadas esféricas, conforme proposto em Josué et al. [2018a]. No entanto, para a definição de uma animação, optou-se pelo suporte apenas ao uso de coordenadas esféricas. Isto porque, o sistema de coordenadas definido através de eixos com posicionamento pré-estabelecidos limita os efeitos de animação.

A Listagem 4.5 mostra um exemplo de animação de um efeito sensorial. No exemplo é definido um efeito sensorial de luz posicionado a 35 graus polar e 45 graus azimutal. Para especificar a animação do efeito sensorial é criado um elo (*link*) e um conector (*causalConnector*) que são ativados quando a apresentação da mídia do efeito sensorial é iniciada e gera uma ação de atribuição para a propriedade de posicionamento azimutal do objeto de mídia. A atribuição definirá o valor de 135 à propriedade azimutal, inicializada com o valor de 45, com a duração (*duration*) de 3 segundos e o incremento (*by*) de 45, ou seja, a cada segundo será incrementado 45 no valor da propriedade até chegar, após 3 segundos, ao valor especificado de 135. Desta maneira, é representada a animação do efeito sensorial iniciada no canto superior esquerdo do

¹Utilizou-se como base a implementação do *middleware* Ginga disponível em <https://github.com/TeleMidia/ginga>

ambiente da apresentação e transladando para o canto superior direito.

```

1 <ncl>
2   <head>
3     <regionBase>
4       <region id="regionSE" polar="35deg" azimuthal="45deg" />
5       <!-- ... -->
6     </regionBase>
7     <descriptorBase>
8       <descriptor id="descSE" region="regionSE" />
9       <!-- ... -->
10    </descriptorBase>
11    <connectorBase>
12      <causalConnector id="onBeginSet_var_duration_by">
13        <connectorParam name="var" />
14        <connectorParam name="duration" />
15        <connectorParam name="by" />
16        <simpleCondition role="onBegin" />
17        <compoundAction operator="seq">
18          <simpleAction role="set" value="var"
19            duration="duration" by="by" />
20        </compoundAction>
21      </causalConnector>
22      <!-- ... -->
23    </connectorBase>
24    <!-- ... -->
25  </head>
26  <body>
27    <!-- ... -->
28    <media id="mediaSE" type="application/x-sensory-effect-LightType"
29      descriptor="descSE">
30      <property name="azimuthal" />
31    </media>
32    <!-- ... -->
33    <link id="linkSE" xconnector="onBeginSet_var_duration_by">
34      <bind role="onBegin" component="mediaSE" />
35      <bind role="set" component="mediaSE" interface="azimuthal">
36        <bindParam name="var" value="135" />
37        <bindParam name="duration" value="3s" />
38        <bindParam name="by" value="45" />
39      </bind>
40    </link>
41    <!-- ... -->
42  </body>
43 </ncl>

```

Listing 4.5: Exemplo de animação de efeitos sensoriais.

Conforme descrito na Seção 2.4, a classe `PlayerAnimator` é responsável por controlar a

atualização dos valores de propriedades dos objetos de mídia. Para dar suporte à animação de efeitos sensoriais, foi necessário realizar a sobrecarga do método *update* do animador (ver Figura 16) para reconhecer como parâmetros as propriedades de posicionamento de efeitos sensoriais, *polar* e *azimuthal*. Nos demais tipos de mídia a atualização das propriedades é feita cada vez que uma nova unidade de informação da mídia é renderizada (*e.g.*, um *frame*). Como os efeitos sensoriais não são renderizados como *frames* em uma tela, a cada passagem do tempo da apresentação, é verificado se uma propriedade deve ser atualizada dado o valor do incremento e, se necessário, o valor da propriedade é atualizado. Quando o valor de uma propriedade é atualizado, duas mensagens contendo a descrição SEM são enviadas para os atuadores, uma contendo o comando para desativar os efeitos sensoriais na posição antiga, e outra para ativar o efeito na nova posição.

4.4 Evento de preparação

Em uma aplicação mulsemídia, o autor pode especificar relacionamentos de sincronização entre objetos de mídia e efeitos sensoriais. A manutenção destas relações de sincronização durante a reprodução da aplicação é de fundamental importância para manter a QoE (do inglês *Quality of Experience*) do usuário que está consumindo o conteúdo.

Com o objetivo de tentar reduzir ou até mesmo evitar atrasos durante a apresentação da aplicação mulsemídia, os autores de Josué et al. [2018b] propõem a incorporação de um novo evento na linguagem NCL, denominado evento de preparação. Considerando objetos de mídia tradicionais, o evento de preparação é disparado quando o conteúdo começa a ser buscado no servidor remoto, e tem seu fim quando o conteúdo requisitado for completamente carregado no receptor, ou quando o *buffer* do *player* estiver completo (cheio), sendo considerada a situação que ocorrer primeiro.

De acordo com Josué et al. [2018b], o evento de preparação é genérico o suficiente para ser aplicado tanto a objetos de mídia, quanto a efeitos sensoriais. A preparação de um efeito sensorial deve considerar diferentes fatores como (*i*) o tempo para um comando ser transmitido do formatador para o dispositivo atuador, (*ii*) o tempo para o efeito atingir a intensidade desejada pelo autor, e (*iii*) o tempo para o efeito ser percebido pelo usuário.

Considerando o fator (*i*), em casos onde o formatador necessita instanciar recursos computacionais complexos para transmitir os comandos, como a conexão com atuadores em redes TCP/IP, no caso do protocolo MQTT, esta etapa pode levar a um grande atraso para iniciar a apre-

sentação do efeito sensorial. Neste contexto, este trabalho propõe a instanciação destes recursos durante a etapa de preparação, de maneira que o tempo de preparação não afete a apresentação do conteúdo multimídia. No protótipo desenvolvido utilizando o protocolo MQTT, considerou-se que o evento de preparação de um efeito sensorial tem seu início quando uma a conexão com o *Broker MQTT* é iniciada; e é finalizado quando a conexão é estabelecida. O tratamento dos outros fatores relacionados à preparação de efeito (*ii e iii*) é deixado como trabalho futuro.

Para a integração do evento de preparação proposto em Josué et al. [2018b], a classe `PlayerSE` sobrescreve o método `startPreparation` da classe `Player` (ver Figura 16) que dá suporte ao evento `Preparation`. Quando a preparação é iniciada o método `startPreparation` é invocado no `PlayerSE` que estabelece a conexão com os dispositivos atuadores através do método `connect` da interface `Protocol`. A conexão deve ser estabelecida de maneira assíncrona ou de forma que não bloqueie a execução da `thread` principal do `middleware` Ginga para não interferir na sincronização das outras mídias. Após o estabelecimento da conexão o `player` é notificado através do método `setPrepared`, indicando a transição no estado da preparação e o evento é encerrado.

4.5 Caso de uso

Esta seção apresenta um caso de uso de uma aplicação NCL com efeitos sensoriais, a fim de apresentar a extensão do `middleware` Ginga proposta neste trabalho². A aplicação NCL utiliza efeitos sensoriais sincronizados com conteúdo audiovisual, o evento de preparação aplicado a efeitos sensoriais e animação do posicionamento de um efeito sensorial. Como foi dito anteriormente, neste trabalho, consideramos a preparação de um efeito sensorial como a conexão a priori com os dispositivos atuadores responsáveis pela renderização do efeito.



Figura 18: Visão geral da aplicação NCL.

²A extensão aqui proposta está disponível em <https://github.com/GPMM/ginga/tree/sensory-effects>

A Figura 18 apresenta uma visão geral da aplicação. Ela é composta por um vídeo principal que apresenta o movimento do sol no céu ao longo do dia, sincronizado com um efeito de calor. A aplicação é iniciada com um vídeo de espera, indicando o carregamento, que dura o tempo da preparação do efeito sensorial. Após a preparação, o vídeo de carregamento é encerrado e o vídeo principal é iniciado juntamente com um efeito de calor. A posição do efeito de calor é alterada ao longo da aplicação de forma a simular o movimento do sol ao longo do dia. A Listagem 4.6 apresenta trechos do código NCL dessa aplicação.

```

1 <head>
2 ...
3 <region id="rScreen" width="100%" height="100%" />
4 <region id="rHeat" polar="30deg" azimuthal="45deg" width="10deg" height="10deg" />
5
6 ...
7 <descriptor id="dHeat" region="rHeat">
8   <descriptorParam name="intensity" value="5.0" />
9 </descriptor>
10 <descriptor id="dLoading" region="rScreen" />
11 <descriptor id="dVideo" region="rScreen" />
12 ...
13 <causalConnector id="onBeginStartPrep">
14   <simpleCondition role="onBegin" />
15   <simpleAction role="startPreparation" />
16 </causalConnector>
17
18 <causalConnector id="onEndPrepStopStart">
19   <simpleCondition role="onEndPreparation" />
20   <simpleAction role="stop" />
21   <simpleAction role="start" />
22 </causalConnector>
23
24 <causalConnector id="onBeginSet">
25   <connectorParam name="var" />
26   <connectorParam name="duration" />
27   <connectorParam name="by" />
28   <simpleCondition role="onBegin" />
29   <compoundAction operator="seq">
30     <simpleAction role="set" value="var" duration="duration" by="by" />
31   </compoundAction>
32 </causalConnector>
33 ...
34 </head>
35 <body>
36 <port id="pinit" component="loading" />
37
38 <media id="loading" src="loading.ogv" descriptor="dLoading" />
39

```

```

40 <media id="video" src="sun.mp4" descriptor="dVideo"/>
41
42 <media id="heat" type="application/x-sensory-effect-HeatType" descriptor="dHeat"/>
43
44 <link xconnector="onBeginStartPrep">
45   <bind role="onBegin" component="loading"/>
46   <bind role="startPreparation" component="heat"/>
47 </link>
48
49 <link xconnector="onEndPrepStopStart">
50   <bind role="onEndPreparation" component="heat"/>
51   <bind role="stop" component="loading"/>
52   <bind role="start" component="video"/>
53 </link>
54
55 <link xconnector="onBeginStart">
56   <bind role="onBegin" component="video"/>
57   <bind role="start" component="heat"/>
58 </link>
59
60 <link xconnector="onBeginSet">
61   <bind role="onBegin" component="heat"/>
62   <bind role="set" component="heat" interface="azimuthal">
63     <bindParam name="var" value="135"/>
64     <bindParam name="duration" value="17s"/>
65     <bindParam name="by" value="30"/>
66   </bind>
67 </link>
68 </body>

```

Listing 4.6: Trecho de código NCL com efeitos sensoriais

A reprodução dos efeitos sensoriais na execução dessa aplicação foi simulada usando uma maquete que representa o ambiente de execução e LEDs que simulam cada diferente atuador no ambiente. A maquete foi posicionada de forma a permitir que o conteúdo visual da aplicação pudesse ser visualizado em conjunto com a ativação dos LEDs. A Figura 19 ilustra a montagem da maquete para a simulação da aplicação.

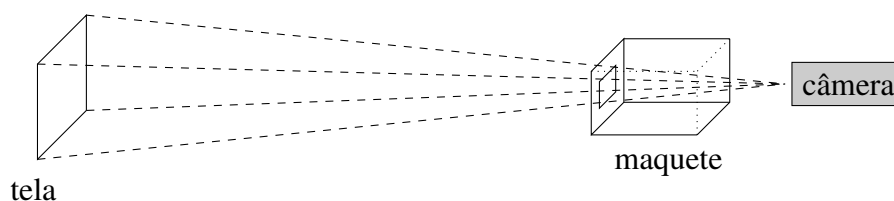


Figura 19: Esquema de simulação da aplicação com efeitos sensoriais

A Figura 20 apresenta o resultado da execução da aplicação. Nela são apresentados diferen-

tes momentos da aplicação. É possível verificar que, conforme o sol se move no vídeo principal da aplicação, os atuadores nas posições correspondentes são ativados³.

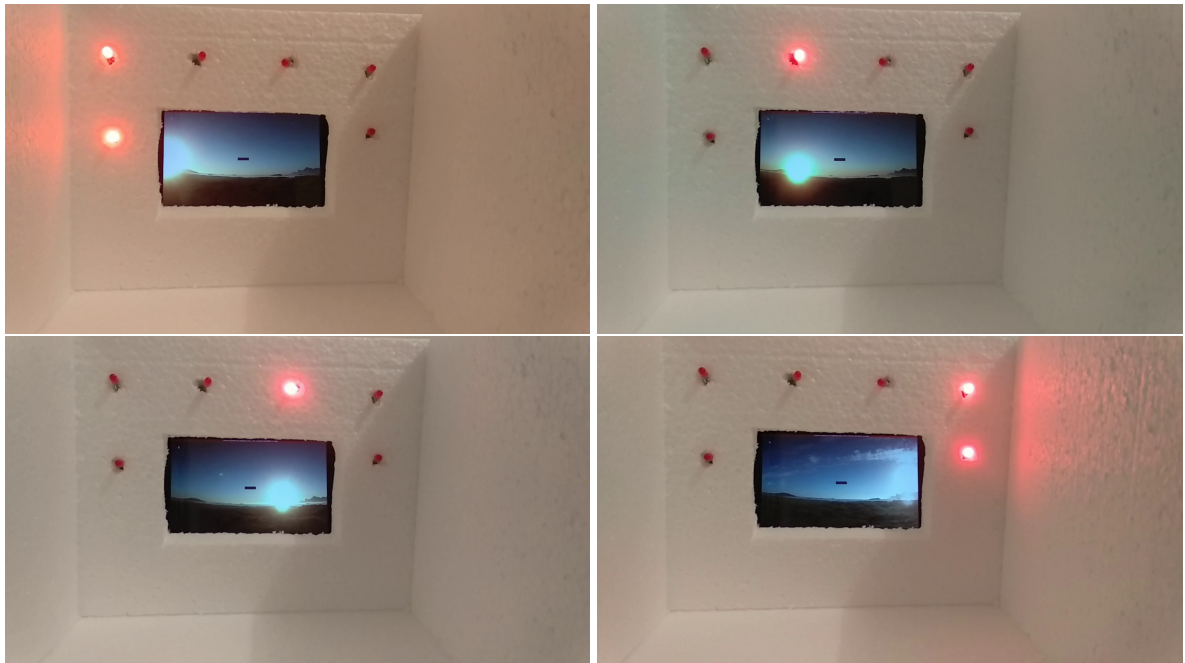


Figura 20: Execução da aplicação com efeitos sensoriais

O exemplo foi executado utilizando o protocolo MQTT conforme discutido na Seção 4.2. Conforme um efeito é executado, o *middleware* Ginga publica a descrição SEM no tópico referente a sua localização, indicando sua ativação ou desativação. O controle dos LEDs que representam os atuadores de calor é feito por um *Raspberry Pi*⁴ executando um cliente MQTT. O *Raspberry Pi*, ao iniciar sua execução, se inscreve nos tópicos referentes ao posicionamento de cada atuador no ambiente de execução. Conforme tais tópicos são atualizados, o *Raspberry Pi* verifica a ação a ser executada e em que atuador. Dessa forma, é possível observar a ativação e desativação dos “atuadores” ao longo do tempo. É importante observar que, a ativação e desativação dos atuadores nesse exemplo é causada pela animação da posição do efeito de calor, conforme definido no código da Listagem 4.6.

³Um vídeo da aplicação sendo executada no esquema aqui apresentado está disponível em <https://bit.ly/2VYPEFI>

⁴Tanto o código da aplicação quanto o código executado pelo *Raspberry Pi* estão disponíveis em <https://github.com/GPMM/WebMedia2019-Example>

Capítulo 5

Conclusão

Aplicações mulsemídia têm sido utilizadas para aumentar a imersão e possibilitar uma melhor qualidade de experiência do usuário que está consumindo o conteúdo. Entretanto, a autoria e execução deste tipo de aplicação não é trivial, uma vez que demanda mecanismos para representar e controlar os efeitos sensoriais, que são renderizados através de dispositivos físicos.

Neste contexto, este trabalho propõe uma extensão a linguagem NCL e ao *middleware* Ginga a fim de desenvolver ferramentas para a autoria e reprodução de aplicações mulsemídia. Para isso, foram criados novos tipos de mídia na linguagem NCL, especificados através do atributo `type` do elemento `<media>`, e a extensão do elemento `<region>`, que atualmente suporta apenas regiões bidimensionais, de modo a permitir a especificação de regiões para efeitos sensoriais utilizando coordenadas esféricas ou as coordenadas definidas pelo padrão MPEG-V. Do ponto de vista do *middleware* Ginga, foi criado um reprodutor de efeitos sensoriais que controla os atuadores durante a apresentação, para isto, são enviadas mensagens contendo uma descrição SEM, de acordo com a SEDL do padrão MPEG-V, para os dispositivos que renderizam os efeitos sensoriais. O envio das mensagens para os atuadores é feito utilizando o protocolo MQTT, no entanto, foi feita uma interface adaptável de forma que seja fácil implementar outros protocolos de comunicação futuramente para suportar a integração com mais dispositivos.

Dado que o tempo para a transmissão das mensagens do *middleware* Ginga para os atuadores pode gerar atrasos e impactar na sincronização com as outras mídia durante uma apresentação, comprometendo a qualidade da experiência dos usuários, este trabalho utiliza os eventos de preparação da linguagem NCL com o objetivo de reduzir estes efeitos. Durante a etapa de preparação, são instanciados os recursos computacionais para a comunicação com os atuadores, no caso do protocolo MQTT utilizado neste trabalho, a conexão TCP/IP, e é encerrado quando estes recursos são obtidos. Desta forma, o tempo de preparação de um efeito sensorial não afeta a apresentação do conteúdo multimídia.

Outra contribuição importante do trabalho foi o suporte à animação de efeitos sensoriais. É possível definir uma animação em NCL por meio do evento de atribuição, modificando as propriedades de posicionamento ou tamanho de um objeto de mídia durante a apresentação. Para

suportar a animação de efeitos sensoriais o evento de atribuição deve ser capaz de realizar modificações nas propriedades de posicionamento dos objetos de mídia de efeitos sensoriais. Desta forma, quando estas propriedades são modificadas, são enviadas mensagens para desativar o efeito sensorial da posição antiga e ativar os efeitos na nova posição.

Este trabalho é focado na reprodução de efeitos sensoriais através de dispositivos atuadores que renderizam os efeitos de acordo com a apresentação mulsemídia. Outro aspecto importante de aplicações mulsemídia é a captação de informações de dispositivos sensores e a modificação do estado da apresentação baseado nesta informação, é um trabalho futuro possibilitar o recebimento de mensagens de dispositivos sensores e disparar eventos NCL de acordo com estas mensagens.

Referências Bibliográficas

- Banks, A. and Gupta, R. (2014). Mqtt version 3.1. 1. *OASIS standard*, 29.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Ghinea, G., Timmerer, C., Lin, W., and Gulliver, S. R. (2014). Mulsemmedia: State of the art, perspectives, and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(1s):17.
- Guedes, A. L., de Cunha, M., Fuks, H., Colcher, S., and Barbosa, S. D. J. (2016). Using ncl to synchronize media objects, sensors and actuators. In *Proceedings of 1st International Workshop on Synchronism of Things (WSOT)*, Webmedia '16.
- ISO/IEC (2016a). Information technology – media context and control – part 1: Architecture. ISO/IEC 23005-1:2016, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC (2016b). Information technology – media context and control – part 2: Control information. ISO/IEC 23005-2:2016, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC (2016c). Information technology – media context and control – part 3: Sensory information. ISO/IEC 23005-3:2016, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC (2016d). Information technology – media context and control – part 4: Virtual world object characteristics. ISO/IEC 23005-4:2016, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC (2016e). Information technology – media context and control – part 5: Data format for interaction devices. ISO/IEC 23005-5:2016, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.

- ISO/IEC (2016f). Information technology – media context and control – part 6: Common types and tools. ISO/IEC 23005-6:2016, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC (2016g). Information technology – media context and control – part 7: Reference software and conformance. ISO/IEC 23005-7:2016, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- Josué, M., Abreu, R., Barreto, F., Mattos, D., Amorim, G., dos Santos, J., and Muchaluat-Saade, D. (2018a). Modeling sensory effects as first-class entities in multimedia applications. In *Proceedings of the 9th ACM Multimedia Systems Conference, MMSys '18*, pages 225–236, New York, NY, USA. ACM.
- Josué, M., Muchaluat-Saade, D., and Moreno, M. (2018b). Preparation of media object presentation and sensory effect rendering in mulsemedia applications. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, pages 45–52. ACM.
- Kim, S. and Han, J. (2014). Text of white paper on mpeg-v. In *San Jose, USA. MPEG Group Meeting, ISO/IEC JTC*, volume 1.
- Lima, G. F. (2018). Architecture and internal api. <http://www.telemedia.puc-rio.br/~gflima/misc/ginga/intapi.html>. Online; accessed 27 November 2018.
- Pereira, F. and Burnett, I. (2003). Universal multimedia experiences for tomorrow. *IEEE Signal Processing Magazine*, 20(2):63–73.
- Saleme, E. a. B. and Santos, C. A. S. (2015). Playsem: A platform for rendering mulsemedia compatible with mpeg-v. In *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web, WebMedia '15*, pages 145–148, New York, NY, USA. ACM.
- Saleme, E. B., Santos, C. A., and Ghinea, G. (2018). Coping with the challenges of delivering multiple sensorial media. *IEEE MultiMedia*.
- Soares, L. F. G., Moreno, M. F., Costa, R. R., de Salles Soares Neto, C., Moreno, M. F., Batista, C. E. F., Santa'anna, F., Savignon, R., Nogueira, F., Lima, G. F., Lima, B. S., de Albuquerque Azevedo, R. G., de Sousa Júnior, J. G., Araújo, E. C., and Álvaro Veiga dna Felipe Nagato (2009). Nested context language 3.0: Part 13 - ginga-ncl implementors guide v1.0. Technical report, Rio de Janeiro, Brasil.

Soares, L. F. G. and Rodrigues, R. F. (2005). Nested context model 3.0. part 1 – ncm core. Technical report, Rio de Janeiro, Brasil.