

UNIVERSIDADE FEDERAL FLUMINENSE

DOUGLAS PAULO DE MATTOS

**STEVE - Editor Gráfico da Visão Espaço-Temporal  
para Autoria de Documentos Hipermissão**

NITERÓI

2016

UNIVERSIDADE FEDERAL FLUMINENSE

DOUGLAS PAULO DE MATTOS

# **STEVE - Editor Gráfico da Visão Espaço-Temporal para Autoria de Documentos Hipermedia**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Sistemas de Computação

Orientador:

Profa. Dra. Débora Christina Muchaluat Saade

NITERÓI

2016

Douglas Paulo de Mattos

STEVE - Editor Gráfico da Visão Espaço-Temporal para Autoria de Documentos  
Hipermissão

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Sistemas de Computação

Aprovada em Abril de 2016.

BANCA EXAMINADORA

---

Profa. Dra. Débora Christina Muchaluat Saade -  
Orientadora, UFF

---

Prof. Dr. Igor Monteiro Moraes, UFF

---

Profa. Dra. Simone Diniz Junqueira Barbosa, PUC-Rio

Niterói  
2016

*Dedico este trabalho à minha noiva, Talita Ramos da Silva, pelo constante apoio, enorme incentivo e compreensão ao longo do desenvolvimento deste projeto. Também dedico aos meus pais, Solange Paulo de Mattos e Levi Neves de Mattos, pela compreensão e apoio.*



# Agradecimentos

Agradeço à minha noiva, Talita Ramos da Silva, pelo constante apoio, enorme incentivo e compreensão ao longo do desenvolvimento deste projeto. Também dedico aos meus pais, Solange Paulo de Mattos e Levi Neves de Mattos, pela compreensão e apoio.

À Professora Dra. Débora Christina Muchaluat Saade, pela excelente orientação, enorme dedicação e paciência, sempre incentivando para obter melhores resultados.

# Resumo

Aplicações multimídia interativas estão disponíveis em várias plataformas como smartphones, computadores, tablets e TVs digitais. Além disso, a produção de conteúdo multimídia tem aumentado cada vez mais e tem sido incentivada pelo fácil acesso a estes dispositivos. Neste cenário, a criação de aplicações multimídia tem ganhado importância. Existem diversas ferramentas comerciais que permitem a autoria de apresentações multimídia utilizando o paradigma de *timeline* (linha do tempo), para usuários sem conhecimento de programação. Estas ferramentas usualmente produzem um vídeo como saída, não gerando documentos hipermídia especificados em alguma linguagem padrão de autoria multimídia, tal como NCL ou HTML5. Ademais, estas ferramentas herdam as limitações do paradigma de autoria *timeline*.

A fim de permitir e facilitar a autoria de documentos hipermídia por usuários sem conhecimento de linguagens de programação e evitar as limitações do paradigma de *timeline*, este trabalho propõe um modelo de documentos hipermídia e um editor gráfico, baseado neste modelo, para edição da visão espaço-temporal de um documento. O modelo, denominado SIMM (Simple Interactive Multimedia Model), faz uso da representação de sincronização temporal baseada em eventos. A ferramenta proposta é chamada de STEVE, *Spatio-Temporal View Editor*. Ela fornece a edição da visão espaço-temporal de documentos hipermídia e permite aos autores criar relações temporais entre os itens de mídia e definir suas características de apresentação. STEVE suporta a definição de relações de interatividade, onde o autor pode especificar ações que venham ocorrer devido à interação do espectador com algum item de mídia da aplicação. E ainda permite realizar a prévia dos documentos, simulando a execução da aplicação multimídia. Além disso, o editor exporta as aplicações hipermídia para documentos NCL e HTML 5, permitindo que as aplicações sejam executadas em diferentes plataformas. Ele pode ser estendido facilmente para suportar outras linguagens multimídia e é implementado utilizando a tecnologia JavaFX para fornecer uma rica interface gráfica e permitir a execução em distintas plataformas.

**Palavras-chave:** autoria multimídia, editor gráfico, modelo de documentos hipermídia, modelo de sincronização temporal, aplicações multimídia interativas, TV digital, NCL, HTML5

# Abstract

Interactive multimedia applications are available in many platforms such as smartphones, computers, tablets and digital TVs. In addition, the production of multimedia content has been growing increasingly and was facilitated due to easier access to these devices. In this scenario, the creation of multimedia applications has gained importance. There are several commercial tools that allow building multimedia presentations using the timeline paradigm for users with no knowledge of programming. These tools usually produce a video as output, not generating hypermedia documents specified in some standard language multimedia authoring such as NCL or HTML5. Moreover, these tools inherit the timeline authoring paradigm limitations.

In order to allow and facilitate the authoring of hypermedia documents for users with no knowledge of programming languages and avoid the timeline paradigm limitations, this dissertation proposes a hypermedia document model and a graphical editor, which is based on this model, for spatio-temporal view editing of a document. The model, which is named SIMM (Simple Interactive Multimedia Model), makes use of the event-based temporal synchronization representation. The proposed tool is called STEVE, Spatio-Temporal View Editor. STEVE provides the editing of spatio-temporal view of hypermedia documents and allows authors to create temporal relationships among media items and define media presentation characteristics. It supports the definition of interactivity relations, where the author can specify actions that will occur due to the viewer interactions with some application media item. It also allows to visualize the document preview, simulating the multimedia application execution. Besides, STEVE exports hypermedia applications to NCL and HTML 5 documents to accomplish different execution platforms. STEVE can be easily extended to support other multimedia languages and it is implemented using JavaFX technology to provide a rich graphical interface and run on distinct platforms.

**Keywords:** Multimedia authoring, graphical editor, hypermedia document model, temporal synchronization model, interactive multimedia applications, digital TV, NCL, HTML

# Lista de Figuras

2.1	Representação de um documento hipermídia . . . . .	7
2.2	Visão estrutural de um documento . . . . .	7
2.3	Visão de leiaute do documento exemplo . . . . .	7
2.4	Visão temporal do documento exemplo . . . . .	8
2.5	Visão espacial do documento exemplo . . . . .	8
2.6	Nós de composição . . . . .	9
2.7	Timeline . . . . .	11
2.8	Flowchart . . . . .	11
2.9	Rede de Petri . . . . .	12
2.10	Sincronização Hierárquica . . . . .	13
2.11	Sincronização baseada em restrições . . . . .	13
2.12	Sincronização baseada em eventos . . . . .	14
2.13	Máquina de estado de eventos . . . . .	16
3.1	Documento NCL representado no modelo de grafo do Composer, retirado de [45] . . . . .	21
3.2	Nó de conectores no modelo de grafo do Composer, retirado de [45] . . . . .	22
3.3	Visão temporal do Composer, retirado de [45] . . . . .	22
3.4	Definição de elos temporais no Composer, retirado de [45] . . . . .	23
3.5	Definição dos elos de interatividade do Composer, retirado de [45] . . . . .	23
3.6	Simulação de eventos interativos do Composer, retirado de [45] . . . . .	24
3.7	Arquitetura do NEXT . . . . .	25
3.8	<i>Drag-and-drop</i> no NEXT . . . . .	26

3.9	Visão Estrutural do NEXT . . . . .	27
3.10	Visão de Leiaute do NEXT . . . . .	27
3.11	Editor de conectores do NEXT . . . . .	28
3.12	Visão hierárquica do CMIFed, retirado de [68] . . . . .	29
3.13	Visão de canal do CMIFed, retirado de [68] . . . . .	29
3.14	Player do CMIFed, retirado de [68] . . . . .	30
3.15	Visualizador de itens de mídia do FireFly, retirado de [33] . . . . .	31
3.16	Editor interativo de documentos do FireFly, retirado de [33] . . . . .	31
3.17	Visão temporal do GRiNs , retirada de [36] . . . . .	32
3.18	Diferenças entre <i>timeline</i> e <i>eventline</i> , retirada de [37] . . . . .	33
3.19	Visão temporal do LimSee 2 , retirada de [22] . . . . .	34
3.20	Ferramentas de edição de conteúdo multimídia, Apple . . . . .	35
3.21	Ferramentas profissionais de edição de vídeo . . . . .	36
3.22	Nero Video 2016, Nero AG . . . . .	36
4.1	Diagrama de classes do modelo de sincronização espaço-temporal do SIMM . . . . .	42
4.2	Visão temporal de um documento hipermídia . . . . .	43
4.3	Modelo da sincronização temporal . . . . .	45
4.4	Cenário de interatividade representada na visão temporal . . . . .	48
4.5	Modelo das propriedades de apresentação . . . . .	50
4.6	Propriedades das classes <i>PositionProperty</i> e <i>SizeProperty</i> . . . . .	50
4.7	Proporções de tela do modelo, retirada de [10] . . . . .	51
4.8	Propriedade de recorte . . . . .	51
4.9	Modelo da sincronização espacial . . . . .	52
5.1	Diagrama de Fluxo de Dados do STEVE . . . . .	60
5.2	Interface Gráfica do STEVE . . . . .	61
5.3	Repositório de Mídias do STEVE. Formato em Ícones. . . . .	63

5.4	Repositório de Mídias do STEVE. Formato em Lista. . . . .	64
5.5	Alertas no Uso do Repositório . . . . .	65
5.6	Interface Gráfica da Visão Temporal do STEVE . . . . .	66
5.7	Abas para Suportar Inúmeras Cadeias Temporais . . . . .	66
5.8	Adicionando Itens de Mídia à Aplicação Hipermissão . . . . .	68
5.9	Otimização da disposição das mídias na cadeia temporal . . . . .	69
5.10	Alertas na criação de uma relação temporal no STEVE . . . . .	71
5.11	Representações gráficas da relação temporal <i>Equals</i> no STEVE . . . . .	72
5.12	Representação da relação <i>Starts</i> no STEVE . . . . .	73
5.13	Representação da relação Temporal <i>Starts With Delay</i> no STEVE . . . . .	74
5.14	Representação da relação <i>Finishes</i> no STEVE . . . . .	74
5.15	Representação da relação <i>Finishes With Delay</i> no STEVE . . . . .	75
5.16	Representação da relação <i>Meets</i> no STEVE . . . . .	75
5.17	Representação da relação <i>Meets With Delay</i> no STEVE . . . . .	76
5.18	Representação da relação <i>Met By</i> no STEVE . . . . .	76
5.19	Representação da relação <i>Met By With Delay</i> no STEVE . . . . .	76
5.20	Definido a relação <i>During</i> no STEVE . . . . .	78
5.21	Representação da relação <i>Overlaps</i> no STEVE . . . . .	78
5.22	Representação da relação <i>Before</i> no STEVE . . . . .	79
5.23	Visualizando as relações temporais de uma mídia . . . . .	79
5.24	Definindo Nova Mídia Interativa . . . . .	82
5.25	Selecionando a tecla interativa . . . . .	83
5.26	Selecionando mídias a serem paradas . . . . .	84
5.27	Visão temporal da aplicação exemplo . . . . .	85
5.28	Janela de definição da relação de interatividade . . . . .	86
5.29	Alertas de validação da relação de interatividade . . . . .	87

5.30	Representação gráfica da relação de interatividade . . . . .	88
5.31	Simulação da Interatividade . . . . .	90
5.32	Conflito: início e fim já definidos . . . . .	91
5.33	Resultado da criação do elo <i>Starts With Delay</i> . . . . .	92
5.34	Nenhuma mídia selecionada pode ser escrava da relação . . . . .	92
5.35	Conflito: início já definido . . . . .	93
5.36	Conflito: término já definido . . . . .	94
5.37	Conflito: Novo início é maior que o fim definido por outra relação . . . . .	95
5.38	Conflito: Novo término é menor que o início definido por outra relação . . . . .	96
5.39	Definindo uma relação alterando o instante de término da mídia . . . . .	97
5.40	Definindo uma relação alterando o instante de início da mídia . . . . .	97
5.41	Comportamento do STEVE quando uma mídia é arrastada . . . . .	99
5.42	Comportamento do STEVE quando a duração de uma mídia mestre é alterada	102
5.43	Interface gráfica da visão espacial do STEVE . . . . .	103
5.44	Aba INFO de um item de mídia . . . . .	104
5.45	Painéis de propriedades dos tipos imagem, vídeo e texto . . . . .	104
5.46	Painéis de propriedades dos tipos imagem, vídeo e texto . . . . .	105
6.1	Arquitetura do STEVE . . . . .	108
6.2	Pacote <i>stevePane</i> do pacote <i>view</i> . . . . .	110
6.3	Diagrama de classes do pacote <i>repositoryPane</i> . . . . .	111
6.4	Diagrama de classes do pacote <i>temporalViewPane</i> . . . . .	111
6.5	Diagrama de classes do pacote <i>spatialViewPane</i> . . . . .	112
6.6	Diagramas de pacote para suporte à NCL e HTML5 . . . . .	114
6.7	Visão espacial da aplicação exemplo . . . . .	116
6.8	HTG para aplicação apresentada na Figura 6.7 . . . . .	117
6.9	Visão temporal para aplicação da Figura 6.7 . . . . .	119

---

6.10	Especificação da relação Starts no modelo SIMM e em NCL . . . . .	121
6.11	Elos criados para definir a relação Before em NCL . . . . .	122
7.1	Etapa 1 da aplicação hipermídia realizada no teste de usabilidade . . . . .	131
7.2	Etapa 2 da aplicação hipermídia realizada no teste de usabilidade . . . . .	131
7.3	Etapa 3 da aplicação hipermídia realizada no teste de usabilidade . . . . .	132
7.4	Resultado do questionário SUS . . . . .	134
7.5	Pontuação média do questionário específico . . . . .	135



# Lista de Tabelas

2.1	Transições na Máquina de estado . . . . .	17
3.1	Recursos das Ferramentas de Autoria . . . . .	39
4.1	Relações Temporais entre Intervalos de Allen . . . . .	44
4.2	Papéis suportados no modelo hipermídia proposto . . . . .	46
4.3	Relações Temporais do Modelo Hipermídia Proposto . . . . .	47
4.4	Relações Espaciais de Alinhamento do Modelo Hipermídia Proposto . . . . .	54
4.5	Relações Espaciais de Distribuição do Modelo Hipermídia Proposto . . . . .	54
5.1	Relações Temporais Oferecidas pelo STEVE . . . . .	71
5.2	Conflitos na Criação das Relações Temporais Síncronas . . . . .	98
6.1	Plano de apresentação para o HTG da Figura 6.8 . . . . .	118
6.2	Relações Temporais Oferecidas pelo STEVE . . . . .	121
7.1	Recursos das Ferramentas de Autoria . . . . .	129
B.1	Papéis Padrões . . . . .	149

# Lista de Abreviaturas e Siglas

aNa	: API for NCL Authoring;
API	: Application Programming Interface;
DOM	: Document Object Model;
GRiNs	: GRaphical iNterface for SMIL;
HTML	: HyperText Markup Language;
NCL	: Nested Context Language;
NCM	: Nested Context Model;
NEXT	: NCL Editor Supporting XTemplate;
SMIL	: Synchronized Multimedia Integration Language;
STEVE	: Spatio-Temporal View Editor;
SIMM	: Simple Interactive Multimedia Model;
XSLT	: Extensible Stylesheet Language Transformation;

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	2
1.2	Objetivos . . . . .	3
1.3	Contribuições . . . . .	4
1.4	Estrutura da Dissertação . . . . .	5
<b>2</b>	<b>Fundamentação Teórica</b>	<b>6</b>
2.1	Sistemas Hipermídia . . . . .	6
2.2	Modelos Hipermídia . . . . .	8
2.3	Paradigmas de Autoria Multimídia . . . . .	9
2.4	Modelo NCM . . . . .	14
2.4.1	Eventos no Modelo NCM . . . . .	15
2.4.2	Sincronização Temporal . . . . .	16
2.5	Conclusão . . . . .	18
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>19</b>
3.1	Ferramentas Acadêmicas . . . . .	19
3.1.1	Composer . . . . .	19
3.1.2	NEXT . . . . .	24
3.1.3	CMIFed . . . . .	27
3.1.4	FireFly . . . . .	30
3.1.5	GRiNs . . . . .	32

---

3.1.6	Eventline . . . . .	33
3.1.7	LimSee 2 . . . . .	34
3.2	Ferramentas Comerciais . . . . .	34
3.3	Requisitos de um Editor Gráfico para Autoria de Documentos Hipermissão .	37
3.4	Comparação dos Trabalhos Relacionados . . . . .	38
3.5	Conclusão . . . . .	38
<b>4</b>	<b>SIMM - Simple Interactive Multimedia Model</b>	<b>40</b>
4.1	Introdução ao SIMM . . . . .	40
4.2	Entidades . . . . .	41
4.3	Sincronização Temporal . . . . .	44
4.4	Interatividade . . . . .	47
4.5	Características de Apresentação . . . . .	49
4.6	Sincronização Espacial . . . . .	52
4.7	Limitações do Modelo SIMM . . . . .	54
4.8	Conclusão . . . . .	56
<b>5</b>	<b>STEVE</b>	<b>57</b>
5.1	Visão Geral . . . . .	58
5.1.1	Arquitetura . . . . .	59
5.1.2	Interface Gráfica . . . . .	60
5.2	Repositório de Mídias . . . . .	62
5.3	Visão Temporal . . . . .	65
5.3.1	Manipulação das Mídias na Visão Temporal . . . . .	67
5.3.2	Sincronização Temporal dos Itens de Mídia . . . . .	70
5.3.3	Suporte à Interatividade . . . . .	80
5.3.4	Mantendo a Consistência Temporal . . . . .	89

---

5.4	Visão Espacial . . . . .	102
5.5	Conclusão . . . . .	105
<b>6</b>	<b>Implementação</b>	<b>107</b>
6.1	Arquitetura de Software . . . . .	107
6.2	Pacotes da Interface Gráfica . . . . .	109
6.2.1	Pacote do Repositório de Mídias . . . . .	110
6.2.2	Pacote da Visão Temporal . . . . .	111
6.2.3	Pacote da Visão Espacial . . . . .	112
6.3	Suporte à NCL e HTML5 . . . . .	113
6.3.1	Leitura de Documentos NCL . . . . .	114
6.3.2	Exportando para NCL e HTML 5 . . . . .	119
6.4	Conclusão . . . . .	126
<b>7</b>	<b>Avaliação da Ferramenta</b>	<b>127</b>
7.1	Comparação com os Trabalhos Relacionados . . . . .	127
7.2	Teste de Usabilidade . . . . .	129
7.2.1	Metodologia . . . . .	130
7.2.2	Questionários . . . . .	132
7.2.3	Resultados . . . . .	133
7.3	Conclusão . . . . .	136
<b>8</b>	<b>Conclusão</b>	<b>137</b>
8.1	Contribuição . . . . .	137
8.2	Limitações do Editor STEVE . . . . .	139
8.3	Trabalhos Futuros . . . . .	140
	<b>Referências</b>	<b>141</b>

---

<b>Apêndice A - Diagrama de Classe Completo do Modelo do STEVE</b>	<b>146</b>
<b>Apêndice B - A Linguagem NCL</b>	<b>148</b>
<b>Apêndice C - Base de Conectores</b>	<b>152</b>

# Capítulo 1

## Introdução

Aplicações hipermídia estão presentes em diversas plataformas como computadores, smartphones, tablets e TVs digitais. A facilidade de acesso a estes dispositivos tem aumentado não só a produção de conteúdo multimídia como também tem tornado a criação de aplicações hipermídia cada vez mais relevante. Diante deste cenário, diversas ferramentas comerciais tem sido disponibilizadas para diferentes plataformas, a fim de permitir a construção de apresentações multimídia mesmo para usuários que não possuem conhecimento de programação.

Estes editores comerciais fornecem uma linha do tempo onde são exibidos todos os itens de mídia da aplicação multimídia. Dessa maneira, eles visam facilitar o entendimento do autor sobre a ordem de apresentação dos itens de mídia da aplicação, utilizando o paradigma de sincronização baseado em *timeline*. O qual oferece uma interface bem intuitiva e fácil de ser compreendida pelos autores, visto que esse modelo é bem aceito no âmbito comercial.

Além disso, essas ferramentas permitem que os usuários, através de uma interface gráfica, editem seus documentos multimídia arrastando os componentes de mídia ao longo do eixo do tempo exibido pela ferramenta e modifique também as propriedades de apresentação desses itens de mídia. De forma conjunta com a visão temporal das apresentações multimídia, os editores ainda oferecem a visão espacial a fim de permitir que os autores verifiquem como os itens de mídia de sua aplicação multimídia estão relacionados espacialmente e como eles são apresentados durante a execução da aplicação.

A integração da visão temporal com a espacial permite que os autores analisem as características de apresentação dos itens de mídia em um determinado instante de tempo da execução da aplicação multimídia. Em outras palavras, a visão espacial é mostrada

para cada instante na linha do tempo. Desta maneira, os autores estão aptos a simular suas aplicações verificando a visão espacial ao longo do tempo. Esta prévia da aplicação não permite que o espectador interaja com os itens de mídia, uma vez que ela é uma simulação e não a execução da aplicação propriamente dita.

## 1.1 Motivação

Apesar da popularidade do paradigma *timeline* no âmbito comercial, visto que essa forma de implementação de editores gráficos permite que usuários sem conhecimento em programação consigam criar suas próprias aplicações multimídia, estas ferramentas possuem diversas limitações intrínsecas ao paradigma *timeline*.

Dado que os itens de mídia do documento são colocados, diretamente, no tempo, esta abordagem é apropriada apenas quando o início e término das mídias são conhecidos a priori ou especificados explicitamente. Nos casos em que existem eventos assíncronos a serem definidos, testes de condição ou interatividade, este paradigma se torna incapaz de representá-los.

Além dessa desvantagem, o paradigma de *timeline* não permite especificar relações de sincronização temporal causal entre os itens de mídia, o que pode dificultar a edição do documento. Por exemplo, considere uma aplicação multimídia, onde uma imagem inicia o documento e termina após 10 segundos. Logo em seguida, um vídeo tem seu evento de apresentação iniciado. Com este vídeo, diversos arquivos de texto representando legendas são sincronizados. Suponha que o autor deseja aumentar a duração da imagem e iniciar o vídeo imediatamente após esse novo instante de término da imagem. Dessa forma, todos os itens de legenda perderiam a sincronização temporal com o vídeo, já que nenhuma relação direta entre o vídeo e as legendas existe, pois o paradigma de *timeline* não permite essa especificação. Neste caso, o autor teria que, novamente, ordenar todos os arquivos de legenda com vídeo.

Ademais, adaptação de conteúdo e composições atemporais não podem ser expressas diretamente na linha do tempo. Outro obstáculo que estas ferramentas comerciais enfrentam é o fato de que elas não produzem documentos hipermídia especificados em alguma linguagem padrão de autoria multimídia. Geralmente, elas codificam a aplicação multimídia em um formato final de vídeo.

Um outro paradigma de autoria, o qual é bastante utilizada no meio acadêmico, é baseado na ocorrência de eventos durante a execução da aplicação multimídia. O modelo



baseado em eventos é bastante expressivo [58] na definição dos relacionamentos temporais entre as mídias, conforme afirmado em [32]. Um modelo de documento hipermídia é dito com maior expressividade que outro quando ele é capaz de representar mais cenários temporais complexos. Além disso, esta abordagem trata facilmente os eventos assíncronos como a interatividade e os testes de variável.

Um exemplo de modelo hipermídia que faz uso deste paradigma é o NCM (Nested Context Model) [65]. Este modelo é muito expressivo, assim uma ferramenta de autoria baseada diretamente neste modelo exigiria do usuário muito conhecimento sobre suas entidades, dificultando o uso da ferramenta, como acontece com as ferramentas de autoria Composer [46] e NEXT [46]. Assim, uma ferramenta fundamentada em um modelo mais simples baseado em eventos facilitaria bastante a autoria de aplicações hipermídia, principalmente para usuários sem conhecimento no modelo ou linguagem de autoria.

## 1.2 Objetivos

A fim de permitir e facilitar a autoria de documentos hipermídia para usuários com nenhum conhecimento em linguagens de programação e evitar as limitações do paradigma de autoria *timeline*, esta dissertação de mestrado propõe um modelo de documento hipermídia e um editor gráfico da visão espaço-temporal baseado neste modelo.

O modelo proposto utiliza a representação de sincronização temporal baseada em eventos e é denominado de SIMM (Simple Interactive Multimedia Model). Ele especifica os relacionamentos temporais e espaciais entre os itens de mídia de acordo com os eventos ocorridos na apresentação. Ele é uma simplificação do modelo NCM para facilitar a autoria de documentos hipermídia, utilizando-o na implementação de um editor gráfico baseado em visão temporal. Desta forma, o modelo proposto evita as limitações do paradigma *timeline*, permitindo definir interatividade e criar relações de sincronização temporal causais entre os itens de mídia.

Além disso, esta dissertação apresenta um editor gráfico da visão espaço-temporal para autoria de documentos hipermídia baseado no modelo proposto. A ferramenta, chamada STEVE (*Spatio-Temporal View Editor*), fornece a edição da visão espaço-temporal de documentos multimídia interativos e permite aos autores criar relações temporais entre os itens de mídia e definir as características de apresentação dos mesmos. Além disso, a ferramenta proposta suporta a definição de interações do espectador e a simulação destes eventos assíncronos, oferecendo a prévia dos documentos hipermídia.

STEVE exporta as aplicações multimídia interativas criadas na ferramenta para documentos NCL e HTML 5. A primeira linguagem de autoria multimídia permite executar as aplicações criadas usando a ferramenta STEVE no Sistema Brasileiro de Televisão Digital (SBTVD) [6] e em serviços IPTV [8] no padrão H.761. No último relatório da Pesquisa Brasileira de Mídia 2015 [17], realizada pela Secretaria de Comunicação Social da Presidência da República sobre os hábitos de consumo de mídia pela população brasileira, foi divulgado que 95% dos entrevistados afirmaram ver TV, dos quais 73% assistem diariamente, valor maior quando comparado à pesquisa do ano de 2014. Além disso, 90% das TVs digitais distribuídas no país devem suportar o middleware GINGA [6] do SBTVD, conforme a alteração do Processo Produtivo Básico (PPB) publicada pelo Ministério da Ciência, Tecnologia e Inovação [12]. Assim, somando-se estes dois fatos, a disponibilização de aplicações hipermídia em NCL e o incentivo ao seu uso é de grande importância, visto que um enorme número de usuários é atingido pelo SBTVD.

Em relação à exportação em HTML 5, STEVE permite que as aplicações sejam executadas nos navegadores Web e, conseqüentemente, exibidas em diversas plataformas como smartphones, tablets e computadores. A ferramenta pode ser facilmente estendida para suportar outras linguagens multimídia e é implementada utilizando a tecnologia JavaFX para fornecer uma rica interface gráfica e executar em distintas plataformas.

## 1.3 Contribuições

Esta dissertação tem como principais contribuições o modelo de documento hipermídia chamado SIMM (Simple Interactive Multimedia Model) e o editor gráfico da visão espaço-temporal denominado STEVE (Spatio-Temporal View Editor) para permitir e facilitar a autoria de documentos hipermídia mesmo por autores que não possuem conhecimento em linguagens de autoria multimídia ou em modelos hipermídia. O modelo proposto é baseado em eventos, sendo expressivo na definição dos relacionamentos temporais tratando a interatividade facilmente.

STEVE oferece a edição da visão espaço-temporal de documentos hipermídia permitindo aos autores definir relações entre os itens de mídia e suas características de apresentação. Além disso, suporta a definição de interatividade e a simulação destes eventos assíncronos. Para permitir que as aplicações criadas na ferramenta sejam executadas em diferentes plataformas, STEVE as exporta para documentos NCL e HTML 5. Ademais, a arquitetura da ferramenta permite que a mesma seja estendida facilmente para

oferecer a exportação para outras linguagens multimídia. Implementada em JavaFX, a ferramenta proposta pode ser executada em diferentes plataformas como computadores e smartphones.

## 1.4 Estrutura da Dissertação

Esta dissertação é estruturada da seguinte maneira. O Capítulo 2 apresenta conceitos básicos sobre os paradigmas de sincronização temporal utilizados pelas ferramentas de autoria multimídia e introduz o modelo NCM. No Capítulo 3, os trabalhos relacionados são apresentados, discutindo sobre ferramentas de autoria multimídia pesquisadas tanto no meio acadêmico quanto comercial. O Capítulo 4 discute o modelo hipermídia proposto. O Capítulo 5 apresenta o editor gráfico da visão espaço-temporal, STEVE, proposto para autoria de documentos hipermídia. A implementação da ferramenta STEVE baseada no modelo proposto é discutida no Capítulo 6, apresentando sua arquitetura de software e seus diagramas de classe e pacotes. O Capítulo 7 realiza a avaliação da ferramenta através de testes de usabilidade e da análise comparativa com os trabalhos relacionados. O Capítulo 8 conclui a dissertação apresentando as contribuições, limitações e trabalhos futuros.

# Capítulo 2

## Fundamentação Teórica

Neste capítulo são apresentados alguns conceitos sobre sistemas e modelos hipermídia nas Seções 2.1 e 2.2 respectivamente. Na Seção 2.3, os paradigmas de autoria multimídia são analisados, discutindo suas vantagens e desvantagens. A Seção 2.4 apresenta o modelo NCM, no qual o modelo hipermídia proposto, SIMM, é baseado.

### 2.1 Sistemas Hipermídia

Sistemas multimídia manipulam mídias discretas ou contínuas, digitalmente, relacionando-as de forma a gerar uma sequência de apresentação das mídias e também definindo de que modo elas são exibidas. Já sistemas hipermídia, além de lidar com mídias discretas ou contínuas como os sistemas multimídia, tratam a interação de usuários, isto é, alteram o fluxo da apresentação dos documentos hipermídia (multimídia interativos) de acordo com a ocorrência da interatividade.

Os documentos hipermídia são compostos por objetos de mídia que representam o conteúdo do documento (textos, imagens, vídeos, áudios etc.) e os relacionamentos entre estes objetos, como demonstrado na Figura 2.1. Além disso, eles possuem diferentes visões que auxiliam o autor a compreendê-los.

A primeira delas é a visão estrutural que mostra como os nós estão relacionados. A Figura 2.2 exibe a visão estrutural de um documento simples.

Outra visão é a de leiaute, que representa as regiões onde as mídias do documento são apresentadas. É importante notar que esta visão apresenta as posições e dimensões iniciais da mídia na tela do dispositivo de exibição, visto que ao longo da execução do documento, modificações podem ser feitas nas propriedades das mídias. Na Figura 2.3, é

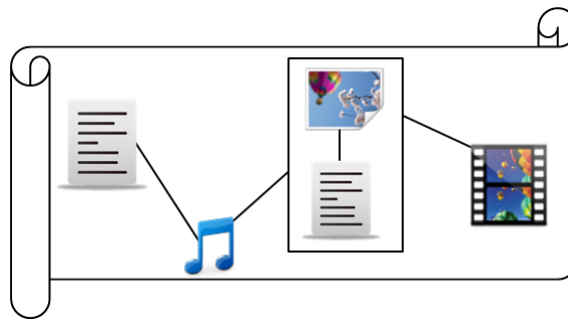


Figura 2.1: Representação de um documento hipermídia

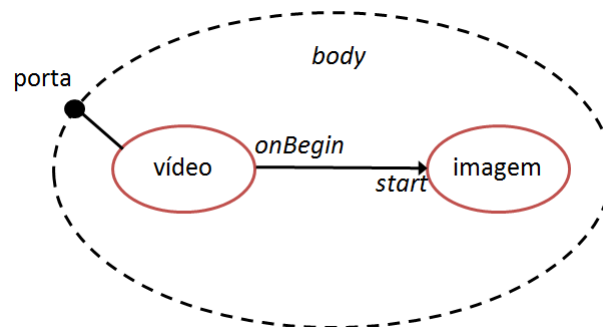


Figura 2.2: Visão estrutural de um documento

mostrada a visão de leiaute onde a região do vídeo ocupa toda a tela da TV.

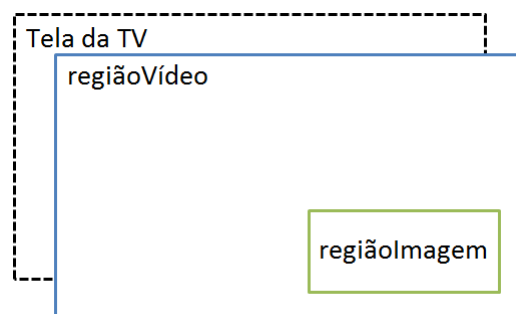


Figura 2.3: Visão de leiaute do documento exemplo

A visão temporal mostra, numa linha do tempo, a ordem em que as mídias do documento são apresentadas e sua duração. Nesta visão, a interatividade é difícil de ser representada, pois depende da interação do usuário com a aplicação hipermídia e não tem um instante predeterminado para acontecer. Sendo assim, os instantes dos objetos de mídia que derivam do elo de interatividade não são determinados a priori. Desta forma, existem diferentes formas de representar a interatividade nesta visão. A Figura 2.4 exibe a visão temporal do documento.

Uma outra visão é a espacial. Ela apresenta como o documento é apresentado num determinado instante de tempo. Isto é, exibe onde e como as mídias são apresentadas nos

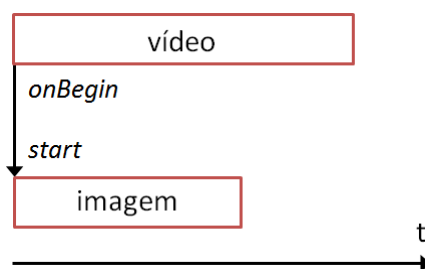


Figura 2.4: Visão temporal do documento exemplo

instantes especificados. A visão espacial do documento é exibida na Figura 2.5.

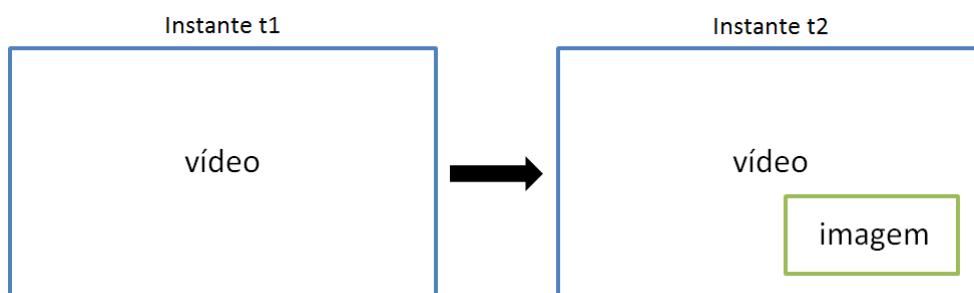


Figura 2.5: Visão espacial do documento exemplo

## 2.2 Modelos Hiperímia

Os componentes de um documento hiperímia apresentados na seção anterior precisam ser expressados através de um modelo conceitual hiperímia. Assim, tais componentes são representados pelas entidades do modelo, sendo as principais, os nós, elos e nós de composição.

Os nós representam os objetos de míia. Cada nó pode possuir âncoras que permitem criar relacionamentos entre as míias do documento. Estas âncoras definem trechos do conteúdo da míia que podem participar de um relacionamento com outra âncora de diferentes nós. O segmento definido por uma âncora depende do tipo de míia, ou seja, se o conteúdo de uma míia é um vídeo, seu trecho pode ser especificado através de um intervalo de tempo ou ainda pelo quadro inicial e final do trecho. No caso de áudio, também pode ser definido por um intervalo de tempo e, além disso, pela amostra inicial e final. Para imagem, um conjunto de pixels formando uma área fechada da mesma (área retangular definido por quatro coordenadas) define o trecho. E por fim, míias do tipo texto têm o segmento especificado através da posição inicial e final no texto. Todas as âncoras de um nó definem sua interface.

Os elos de um documento hiperímia são utilizados para relacionar as âncoras (ponto

de interface) de nós distintos. Estes elos permitem representar os relacionamentos de sincronização, os quais definem o posicionamento espacial e temporal dos objetos. Estes elos podem ter como origens e destinos diversos nós, sendo um elo multiponto.

Os nós de composição possibilitam o agrupamento de diversos nós somente ou também de diferentes nós e elos. Como os nós individuais, um nó de composição pode ter âncoras para permitir o relacionamento diretamente com nós de composição, representadas pelos pontos de interface *a* e *b* simbolizados na Figura 2.6 e também possibilitar o elo entre nós internos (interface *c* na Figura 2.6) de um nó de composição com nós externos no documento. No caso de um ponto de interface que oferece acesso externo a um conteúdo de um nó de composição, a interface é chamada de porta.

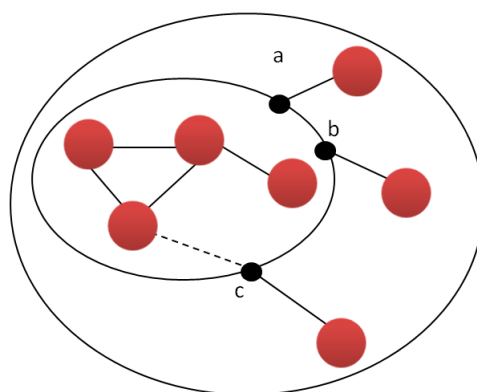


Figura 2.6: Nós de composição

Assim, os nós de composição são utilizados para estruturar um documento agrupando nós e elos, tornando-o mais organizado. Ademais, eles ainda permitem representar relações de sincronização agrupando nós que são apresentados em sequência ou em paralelo, como na linguagem SMIL (Synchronized Multimedia Integration Language) [7].

## 2.3 Paradigmas de Autoria Multimídia

Os relacionamentos temporais entre os nós de um modelo hipermídia podem ser realizados de diferentes maneiras. O tipo de abordagem adotado para a sincronização temporal caracteriza o paradigma de autoria multimídia utilizado pelos sistemas de criação de aplicações hipermídia. Adiante, serão analisados os seguintes paradigmas: scripts, *timeline* (linha do tempo), especificação formal, hierárquica, restrições e baseados em eventos. Esta análise foi baseada nos estudos realizados em [36]. Mais detalhes sobre estes paradigmas podem ser encontradas no Capítulo 4 de [38] e em [48].

### Sincronização baseada em Script

Este modelo faz o uso de programação em *scripts* para definir o relacionamento temporal entre as mídias do documento. Ele é bastante flexível e expressivo, já que usa linguagem imperativa para especificar o comportamento do documento.

Porém, este paradigma exige conhecimento em programação do autor e a visualização de como as mídias estão organizadas na apresentação hipermídia torna-se difícil. Mesmo que a estrutura narrativa da apresentação tenha sido descrita pelo script, a manipulação dessa estrutura em um nível de abstração alto é dificultada.

Outra desvantagem é que a informação de tempo da apresentação também está embutida no código do script, implicando que a aplicação esteja somente disponível em um formato final. Caso alguma plataforma não seja compatível com a linguagem de script utilizada, a aplicação multimídia deixa de ser útil.

No âmbito comercial, *Adobe Flash* [9] é exemplo de ferramenta que faz o uso de scripts para criar apresentações multimídia. Em relação ao meio acadêmico, exemplos de ferramentas que utilizam esse paradigma são: Videobook [57], Harmony [44] e Nsync [28].

### Sincronização baseada em timeline

No caso da sincronização baseada em *timeline*, as mídias são posicionadas diretamente no eixo do tempo e, tipicamente, distribuídas em diferentes trilhas deste eixo, conforme mostrado na Figura 2.7. Desta forma, o comprimento das mídias neste eixo indica diretamente as durações dos objetos de mídia, indicando seus instantes de início e término. Este paradigma é muito intuitivo na criação de apresentações hipermídia como também na sua visualização temporal. Sendo assim este modelo é utilizado por várias ferramentas comerciais como será visto no capítulo de trabalhos relacionados, além de predominar no processo de controle de sincronização do sistema LimSee2 [41].

Entretanto, esse modelo é bastante criticado na literatura por não representar relações temporais diretamente entre os nós que compõem o documento. Desta forma, não é possível garantir a sincronização temporal durante a exibição do documento realizando algum ajuste temporal, caso haja algum atraso na exibição de um nó específico. Outro obstáculo desta abordagem é o fato de que ela é apropriada apenas quando o início e término das mídias são conhecidos a priori ou especificados explicitamente. Assim, nos casos de eventos assíncronos como a interatividade e testes de variável, este paradigma é incapaz de representar estes cenários. Além disso, esta abordagem não expressa a



adaptação de conteúdo e composições atemporais.

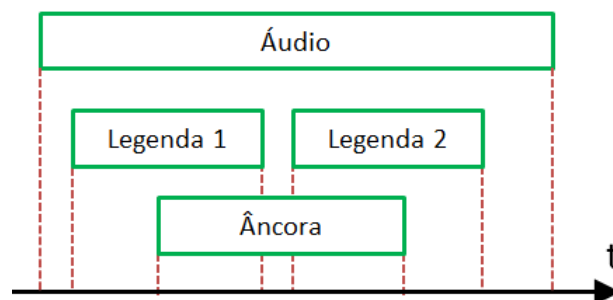


Figura 2.7: Timeline

### Sincronização baseada em especificação formal

Este paradigma utiliza modelos formais para representar os relacionamentos temporais entre as mídias. Como exemplo destes modelos formais, têm-se *Flowcharts* e *Redes de Petri*. O primeiro modelo é um diagrama que representa um processo, descrevendo passo-a-passo, no caso da sincronização temporal em sistemas hipermídia, o comportamento das mídias do documento através de elementos gráficos que representam ações e pontos de decisão conectados por setas, assim, construindo o fluxo de controle da apresentação. Como exemplo do uso deste modelo, tem-se o *Macromedia Authorware* [5] e *FireFly* [33]. Esta representação pode ser vista na Figura 2.8.

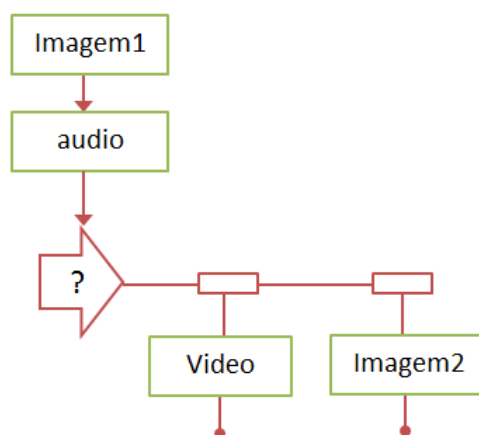


Figura 2.8: Flowchart

As *redes de Petri* são baseadas num grafo dirigido que possui como elementos: os *lugares*, *setas* e *transições*. Cada *lugar* possui duração e uma transição é disparada somente se todos seus *lugares de entrada* contiverem *tokens*. Quando ela é executada, os *tokens* são movidos para seus *lugares de saída* e ficam bloqueados neles no intervalo de tempo

que corresponde a duração dos mesmos. Um exemplo de uso desta notação formal é o *HTSPN* [69]. A Figura 2.9 representa uma rede de Petri.

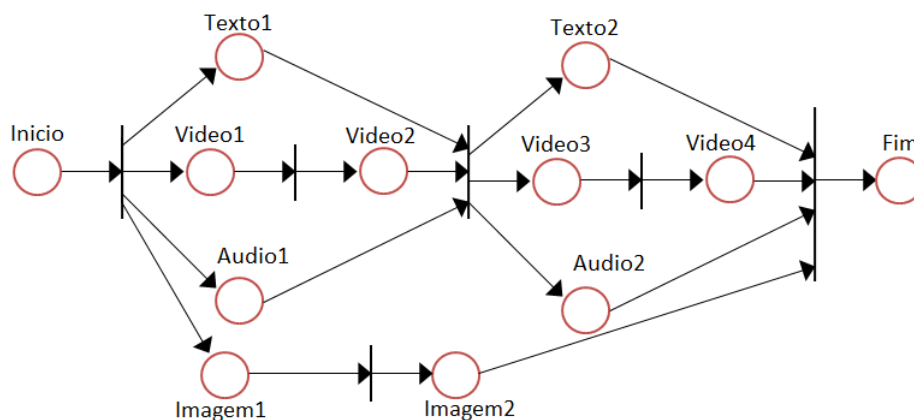


Figura 2.9: Rede de Petri

Este paradigma é bastante expressivo e permite tanto modificações durante a execução da aplicação quanto a verificação formal do comportamento temporal do documento. Uma vantagem dos grafos direcionados é que eles podem fornecer um *framework* para uma análise de tempo formal das apresentações multimídia.

Entretanto, este paradigma requer um conhecimento das notações formais utilizadas e, nos casos de relacionar temporalmente partes de mídias, o modelo formal pode se tornar complexo. Outra desvantagem é a dificuldade presente na especificação dos eventos de interação do usuário com os objetos de mídia.

### Sincronização hierárquica

O modelo de sincronização hierárquico ou estruturado faz o uso de composições que definem a relação temporal entre seus conteúdos. Há dois tipos de composição: sequencial e paralela. Na primeira, todos os nós presentes dentro da composição são apresentados sequencialmente e no caso da segunda, os nós são exibidos simultaneamente no tempo. Na seção de trabalhos relacionados, foram analisados dois sistemas que empregam este paradigma, o *CMIFed* [68] e *GRiNs* [35]. A Figura 2.10 mostra a hierarquia dos nós de um documento.

O modelo permite fazer alterações durante a execução, além de facilitar a autoria de documentos hipermídia e ser bastante expressivo. Ele ainda permite realizar alterações diretamente da estrutura das apresentações. Visto que uma aplicação multimídia pode ser composta de diferentes níveis de composição, o autor está apto a visualizar estes diferentes níveis da aplicação.

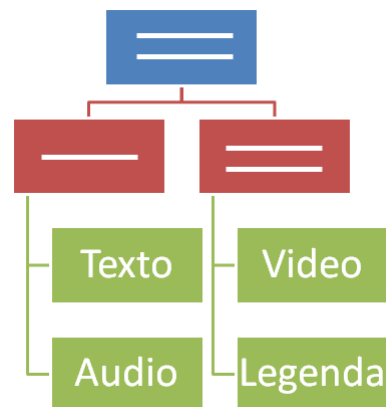


Figura 2.10: Sincronização Hierárquica

Contudo, os eventos de interatividade são dificilmente expressados por este modelo e o relacionamento entre partes de mídias também é dificultado. Além disso, a criação dos componentes de estrutura pode se tornar complexa.

### Sincronização baseada em restrições

Este paradigma utiliza as definições das treze relações básicas de Allen [27] entre instantes ou intervalos de tempo. O modelo facilita a autoria de documentos hipermídia, é bastante expressivo e também permite fazer ajustes durante a execução. Porém, assim como o modelo hierárquico, possui limitações para representar a interatividade e sincronizar entre partes de mídias. O sistema *Madeus* [67] faz uso deste paradigma. A Figura 2.11 exhibe como nós são ordenados no tempo com as relações de [27].

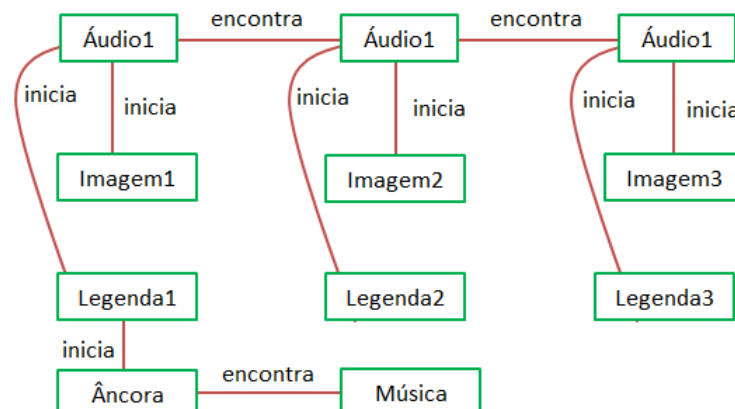


Figura 2.11: Sincronização baseada em restrições

### Sincronização baseada em eventos

Este modelo se baseia na ocorrência de eventos durante a execução do documento para especificar os elos temporais entre os nós. Tais eventos podem ser de apresentação (*start*, *stop* e *pause* de um nó), seleção (interatividade do usuário) ou atribuição (alteração do valor de uma propriedade de um objeto de mídia). Um exemplo de uso deste paradigma é o modelo NCM, utilizado pela linguagem NCL, para expressar documentos hipermídia. Ambos serão vistos nas próximas seções. Na seção de trabalhos relacionados, o Composer [46] e o NEXT [55] fazem uso deste modelo, visto que ele é uma ferramenta para autoria de documentos hipermídia pra TV digital que utiliza a linguagem NCL para especificar as aplicações multimídia interativas. Na Figura 2.12, são mostradas as relações temporais entre os objetos de mídia baseadas em eventos.

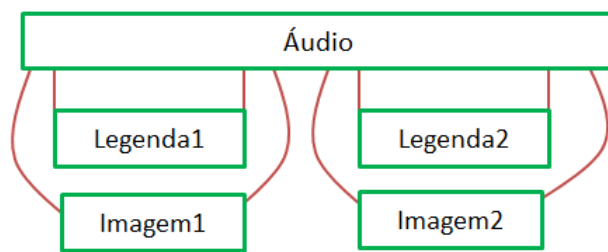


Figura 2.12: Sincronização baseada em eventos

Este modelo é bastante expressivo na definição dos relacionamentos temporais entre as mídias e facilmente trata a interatividade, além de permitir modificações durante a execução do documento. Entretanto, quando uma aplicação hipermídia possui muitos nós e elos temporais a serem especificados entre eles, a autoria pode se tornar complexa, principalmente, para usuários sem conhecimento sobre o modelo utilizado pela ferramenta de autoria.

## 2.4 Modelo NCM

O modelo NCM, (*Nested Context Model*) [65], especifica um documento hipermídia se baseando nas entidades nós e elos. Os nós NCM se dividem em duas classes: nós de mídia ou de conteúdo e nós de composição. Os nós de mídia têm como propriedades básicas o conteúdo da mídia, uma lista de âncoras e um descritor (propriedade opcional). Este último define informações das características de apresentação do item de mídia. Uma âncora pode ser de propriedade ou de conteúdo. No primeiro caso, ela representa propriedades de um objeto de mídia ou de um contexto que podem ser manipuladas por

elos.

O segundo tipo de âncora define um trecho da mídia de acordo com seu tipo. Ou seja, se o conteúdo de uma mídia é um vídeo, seu trecho pode ser especificado através de um intervalo de tempo ou ainda pelo quadro inicial e final do trecho. No caso de áudio, também pode ser definido por um intervalo de tempo e, além disso, pela amostra inicial e final. Para imagem, um conjunto de pixels formando uma área fechada do conteúdo (área retangular definida por quatro coordenadas) define o trecho. E por fim, mídias do tipo texto têm o segmento especificado através da posição inicial e final no texto. Além deste tipo, pode existir uma âncora de propriedade, a qual define atributos associados ao nó.

Os nós de composição, no modelo NCM, são representados pelos nós de contexto e nós *switch*. Os nós de contexto podem incluir nós de mídia, outros de contexto, nós *switch* e elos. Eles visam permitir que seja especificada uma estrutura lógica, hierárquica ou não, para documentos hipermídia. No que diz respeito aos nós *switch*, estes são responsáveis por suportar a adaptação de conteúdo. Eles possuem nós alternativos que são selecionados fazendo o uso de regras especificadas no documento. Isto é, cada regra é avaliada para determinar se o nó de conteúdo associado a ela deve ser apresentado ou não durante a execução do documento.

Os elos NCM fazem referência a um conector e definem um conjunto de *binds*. O conector [56] é a entidade que define a semântica de um relacionamento entre nós de um documento hipermídia, especificando papéis como seus pontos de interface. Os elos, por sua vez, são responsáveis por associar pontos de interface de nós (âncoras e atributos) aos papéis definidos pelo conector através de *binds*. Um conector pode ser reutilizado por vários elos que possuem a mesma semântica.

### 2.4.1 Eventos no Modelo NCM

Um evento multimídia [58] [32] é uma ocorrência no tempo que pode ser instantânea ou ter uma duração mensurável. O modelo NCM define os seguintes tipos de eventos: evento de apresentação, evento de seleção e evento de atribuição. São definidos outros tipos de evento, entretanto, somente esses são relevantes para este trabalho.

O evento de apresentação é relativo à exibição de uma âncora de conteúdo e também de um nó de composição (contexto e *switch*) indicando a apresentação de todo o conteúdo da composição. Em relação ao evento de seleção, este representa a seleção de uma âncora de conteúdo, que está sendo exibida, pelo usuário. Este é o único evento que depende da

interação do espectador. Já o evento de atribuição modifica o valor de uma âncora de propriedade de um objeto de mídia.

Para cada um destes eventos descritos, o NCM define uma máquina de estados indicando em qual estado um determinado evento se encontra. Tais estados podem ser os seguintes: *ocorrendo* (*occurring*), *pausado* (*paused*) e *dormindo* (*sleeping*). Além disso, os eventos possuem algumas propriedades como: *ocorrências* (*occurrences*), que conta o número de vezes que o evento passou do estado *ocorrendo* para o *dormindo* ao longo da apresentação de um documento hipermídia, e *repetições* (*repetitions*), somente para os eventos de atribuição e apresentação, que especifica a quantidade de vezes que o evento deve ocorrer. A Figura 2.13 mostra a máquina de estados de eventos do modelo NCM.

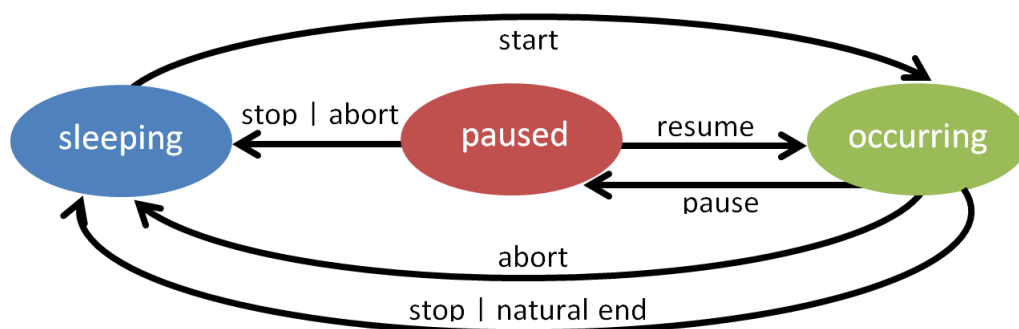


Figura 2.13: Máquina de estado de eventos

É importante ressaltar que a duração de um evento é o tempo durante o qual ele permanece no estado ocorrendo. Tal duração pode ser intrínseca ao objeto de mídia, explicitamente especificada pelo autor do documento ou ainda derivada de um relacionamento.

## 2.4.2 Sincronização Temporal

A sincronização temporal no modelo NCM é baseada em eventos que ocorrem durante a apresentação de um documento. As relações definidas pelos conectores NCM podem ter semântica causal ou de restrição. Nos conectores causais, uma condição deve ser satisfeita para que uma ação seja executada. Estes conectores definem papéis (*roles*) que especificam como cada nó associado a eles devem participar na relação causal. Os papéis ainda possuem um identificador, tipo de evento e sua cardinalidade indicando o número mínimo e máximo de nós que podem ser associados ao papel. O tipo de evento (*event type*) corresponde a um dos eventos apresentados na Seção 2.4.1.

Os papéis podem ser de condição (*condition*), ação (*action*) ou avaliação (*assessment*).

Quanto aos papéis de ação (*action roles*), eles são responsáveis por causar uma transição na máquina de estado de um evento. A Tabela 2.1 descreve cada uma das transições disparadas pelas tipos de ações definidos pelo modelo na máquina de estado.

Tabela 2.1: Transições na Máquina de estado

Transição (Causada pela Ação)	Nome da Transição
<i>sleeping</i> $\rightarrow$ <i>occurring</i> ( <i>start</i> )	<i>starts</i>
<i>occurring</i> $\rightarrow$ <i>sleeping</i> ( <i>stop</i> )	<i>stops</i>
<i>occurring</i> $\rightarrow$ <i>sleeping</i> ( <i>abort</i> )	<i>aborts</i>
<i>occurring</i> $\rightarrow$ <i>paused</i> ( <i>pause</i> )	<i>pauses</i>
<i>paused</i> $\rightarrow$ <i>occurring</i> ( <i>resume</i> )	<i>resumes</i>
<i>paused</i> $\rightarrow$ <i>sleeping</i> ( <i>stop</i> )	<i>stops</i>
<i>paused</i> $\rightarrow$ <i>sleeping</i> ( <i>abort</i> )	<i>aborts</i>

As condições capturadas pelos papéis do tipo condição (*condition roles*) devem ser satisfeitas para executar a ação definida pelo conector. Elas definem expressões lógicas de forma a avaliar valores de atributos dos nós e dos eventos, e também verificar estados dos eventos. Após a avaliação da condição, ela retorna um valor booleano. Tanto as condições quanto as ações podem ser simples ou compostas.

Uma condição simples testa uma transição de estado de um evento e os valores de um atributo de um nó ou de um evento. A condição composta é uma expressão lógica que envolve duas ou mais condições simples ou compostas ligadas pelos operadores *and* ou *or*.

Uma ação simples refere-se a um papel do tipo ação. A ação composta contém duas ou mais ações simples ou compostas ligadas pelos operadores *seq* ou *par*. O primeiro indica que as ações da composição são disparadas em uma determinada sequência e o segundo aponta que elas podem ser executadas em qualquer ordem. Além disso pode também ser especificado um *delay* indicando o intervalo de tempo que deve ser obedecido para executar a ação de um elo após a condição definida pelo conector ser satisfeita.

Para exemplificar, considere um elo entre duas mídias (*vídeo* e *áudio*) que especifica que o término do vídeo deve ser sincronizado com o início do áudio. No relacionamento, a condição simples é o término da apresentação do *vídeo*, ou seja, a ocorrência da transição *stops* da máquina de estado do evento de apresentação deste objeto. A ação simples a ser disparada é iniciar a apresentação do objeto *áudio* (transição *starts*).

Os papéis de avaliação permitem construir condições que checam atributos de eventos, o estado de apresentação de um evento e valores de propriedades de nós. Os atributos de eventos correspondem ao *occurrences* ou *repetitions*, os quais indicam o número de vezes

que um evento já ocorreu e que ele deverá ocorrer depois da ocorrência atual respectivamente. O atributo *repetitions* é definido para todos os tipos de eventos, exceto para os de seleção. Analogamente aos papéis de condição, as avaliações também podem ser simples ou compostas utilizando os operadores *and* e *or*.

No que diz respeito aos conectores de restrição, estes definem relações que não envolvem causalidade entre os eventos do documento. Elas apenas estabelecem que a expressão assertiva deve se manter verdadeira durante a apresentação do documento. Uma expressão assertiva realiza a comparação de papéis de avaliação do mesmo tipo ou de um papel de avaliação com um valor, do mesmo tipo, do resultado da avaliação. Esta comparação é feita utilizando os mesmos operadores definidos para as condições simples de um conector causal. Uma assertiva também pode ser composta utilizando os operadores *and* e *or* de forma análoga às condições compostas.

## 2.5 Conclusão

O capítulo apresentou alguns fundamentos de sistemas hipermídia como suas definições e descrição sobre seus elementos. Além disso, foram estudados os modelos hipermídia utilizados para especificar os documentos multimídia e os paradigmas de autoria multimídia utilizados por diferentes sistemas de autoria. Para cada paradigma, o capítulo apresentou as vantagens e desvantagens. Em seguida, o modelo NCM foi discutido apresentando suas principais entidades e seu modelo de sincronização temporal.

O próximo capítulo irá discutir diversos trabalhos relacionados que propõem sistemas de autoria multimídia que utilizam diferentes paradigmas de autoria. Além das ferramentas acadêmicas, o capítulo também citará softwares comerciais para edição de apresentações multimídia.



# Capítulo 3

## Trabalhos Relacionados

A fim de obter informações sobre as ferramentas de autoria de apresentações hipermídia disponíveis no meio acadêmico e comercial, e analisar seus recursos e limitações, diversas ferramentas foram pesquisadas, estudando seus paradigmas de autoria, modelos sincronização e interface gráfica. As ferramentas comerciais citadas neste trabalho são bastante conhecidas na área de criação e edição de apresentações multimídia, como o *Final Cut Pro* [20], *iMovie* [21], *Davinci Resolve 12* [19], *Adobe Premiere* [18] e *Nero Video 2016* [25]. Em relação às ferramentas acadêmicas, os seguintes sistemas foram estudados: Composer [46], NEXT [55], CMIFed [68], Firefly [33], GRiNs [35], Eventline [37] e LimSee 2 [41].

### 3.1 Ferramentas Acadêmicas

#### 3.1.1 Composer

A ferramenta Composer [46] é um ambiente de autoria integrado (IDE) para o desenvolvimento de aplicações NCL para o sistema brasileiro de televisão digital, facilitando e agilizando a criação de aplicações interativas. A ferramenta pode ser adaptada a diversos perfis de autores e oferece suporte a requisitos não funcionais. A arquitetura é baseada em um micronúcleo que permite a troca de informações entre os diferentes módulos que complementam o ambiente. Além do micronúcleo, um modelo central e extensões compõem a arquitetura. Suas extensões são construídas através de *plugins*, que são programas que interagem com o micronúcleo e acrescentam novas funcionalidades à ferramenta.

O Composer foi desenvolvido baseado no editor HyperProp [66], assim ele também oferece ao autor várias visões do documento NCL (estrutural, leiaute, textual e temporal). A principal alteração feita no HyperProp foi em sua visão temporal quanto à sincronização

dos objetos no tempo e a aplicações com interatividade que geram diferentes cadeias temporais. Na visão temporal do HyperProp, só é possível visualizar as cadeias temporais, diferente do Composer, que permite criar e editar os objetos de mídias ao longo do tempo. As modificações feitas no cadeia temporal são refletidas nas outras visões do documento NCL no Composer.

O modelo de dados utilizado para representar a cadeia temporal hipermídia [40], tanto do Composer quanto do HyperProp, consiste em um grafo que é formado por um conjunto de nós que podem ser nós de início, de término ou de conectores; e arestas unidirecionais. Este modelo leva em consideração a máquina de estado de eventos da linguagem NCL versão 3.0. Ou seja, as transições de início (*starts*) e de recomeço (*resumes*) são representadas pelos nós de início. Já as transições de término (*stops*) e de pausa (*pauses*) são representadas pelos nós de término. Ainda neste modelo, as arestas do grafo podem representar os estados ocorrendo (*occurring*) e pausado (*paused*) de eventos de apresentação (*presentation*). O primeiro estado é representado pelas arestas com origem em nós de início e destino em nós de término. E o estado pausado é representado pelas arestas com origem em nós de término e destino em nós de início. Além disso, as arestas podem definir relacionamentos de causalidade representando condições que devem ser satisfeitas para a execução da transição de destino (nó de destino da aresta). Isto é, a transição de origem deve ser executada e depois de um determinado intervalo de tempo especificado na aresta, a transição de destino é executada. Não só os relacionamentos causais mas também os de restrições podem ser representados. Porém estes relacionamentos de restrições não são considerados no contexto de TV digital.

Assim, para cada nó do grafo é possível obter seu instante de execução na cadeia temporal pelo cálculo da soma dos tempos das arestas visitadas desde do vértice inicial pré-definido até o nó desejado. Neste modelo, há pontos que definem outras cadeias temporais hipermídia. Tais momentos indicam a existência de interatividade, onde os tempos a partir destes pontos, para serem calculados, dependem do instante da interação do usuário com a aplicação. Estas cadeias são consideradas cadeias auxiliares e a cadeia formada pelo nó inicial pré-definido é a principal. Quando esses instantes são definidos, as cadeias auxiliares passam a integrar a cadeia principal. A Figura 3.1 apresenta um exemplo da representação de um trecho de documento NCL no modelo de grafo temporal utilizado.

As letras *I* e *T* representam os nós de início e de término respectivamente. E as letras *W*, *W'*, *W''* representam, nas arestas, a duração de exibição dos objetos de mídia *audio1*, *video1* e *ads1* respectivamente. E a letra *S* é o arco que indica um relacionamento

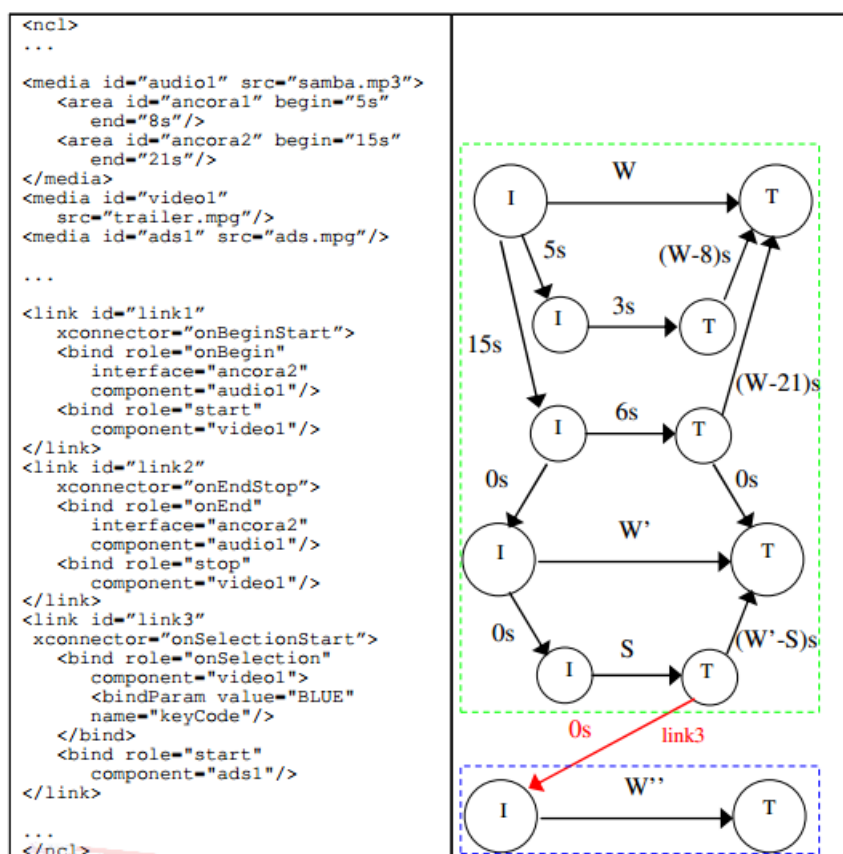


Figura 3.1: Documento NCL representado no modelo de grafo do Composer, retirado de [45]

de interatividade, sendo assim, seu intervalo de tempo é indeterminado. E a partir da aresta *link3* é gerada uma cadeia temporal auxiliar. Cada retângulo pontilhado na figura representa uma cadeia temporal.

Na Figura 3.2, é mostrada a representação de um vértice que representa um conector pelo nó *C*. Este serve como ponto de união em relacionamentos multiponto, no caso o elo *link3*. Ele dispara duas ações simultâneas, a aresta *P* que representa a pausa do objeto *audio1* e a aresta *W''* que inicia a mídia *ads1*. E o elo *link4* dispara o recomeço do objeto *audio1* quando a mídia *ads1* termina sua execução. As arestas de pausa não entram no cálculo do tempo dos nós e elas obtêm seu valor através da diferença dos intervalos de tempo dos nós interligados pela aresta.

Quanto à interface gráfica da visão temporal do Composer, ela possui uma escala de tempo que é alterada de acordo com o zoom utilizado, destacado pelo número 1 na Figura 3.3. Uma barra de status é apontada pelo número 2, que informa o nome da mídia selecionada na cadeia temporal e seu tempo de início e fim de apresentação. E uma barra de tempo indicada pelo número 3. Assinalada pelo número 4, está a representação

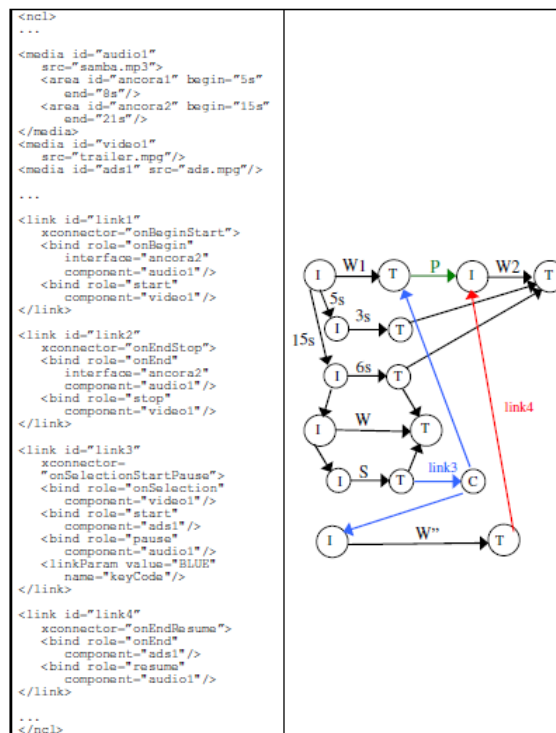


Figura 3.2: Nó de conectores no modelo de grafo do Composer, retirado de [45]

gráfica de uma âncora. No número 5, tem-se a representação de um elo causal entre a âncora apontada pela numeração 4 e o objeto de mídia indicado por 6. Para indicar interatividade, um símbolo é utilizado, indicado pelo número 7 na figura. Um instante de tempo indeterminado é representado por um ícone, destacado pelo número 8.

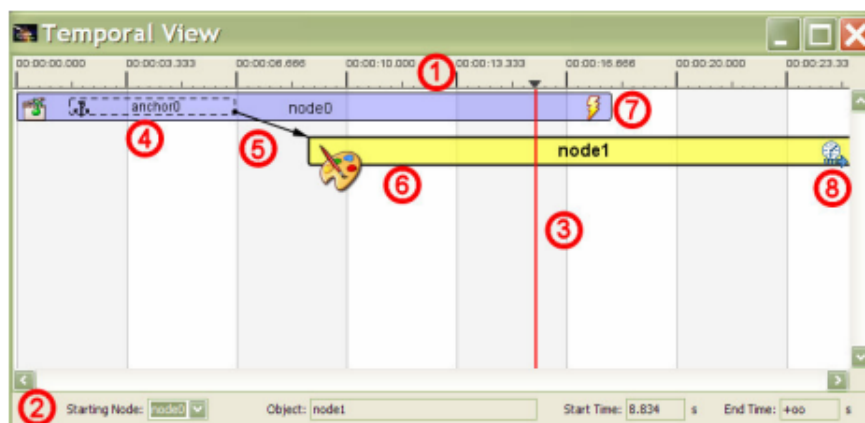


Figura 3.3: Visão temporal do Composer, retirado de [45]

A edição da cadeia temporal hipermídia é feita através de relações de alto nível, assim dividindo os relacionamentos do grafo em dois grupos: os elos de começo/fim e elos de pausa/recomeço. A especificação de relacionamentos temporais é realizada através de caixas de diálogo como a exibida na Figura 3.4.

The 'Create Temporal Relation' dialog box is used to define temporal relationships between media nodes. It features three main sections: 'Media node...', 'Start Time', and 'End Time'. The 'Media node...' section includes a dropdown for '\* Node id' (currently 'video1') and a 'New...' button. The 'Start Time' section includes dropdowns for 'Relative to' (currently 'audio1'), '\* Anchor' (currently 'ancora2'), and '\* Action' (currently 'start'), along with a 'Time' input field and a unit selector (currently 's'). The 'End Time' section includes similar dropdowns for 'Relative to' (currently 'audio1'), '\* Anchor' (currently 'ancora2'), and '\* Action' (currently 'end'), along with a 'Time' input field and a unit selector (currently 's'). At the bottom, there are 'OK', 'Cancel', and 'Help' buttons, and a note '\* Required fields.'.

Figura 3.4: Definição de elos temporais no Composer, retirado de [45]

Os relacionamentos de interatividade também são definidos em caixas de diálogo como mostrado na Figura 3.5, especificando o momento que ela pode ocorrer, a tecla do controle remoto que deve ser pressionada pelo telespectador para interagir e a ação a ser executada após a interatividade. Esta ferramenta da visão temporal só permite a especificação de relações de interatividade um para um, o que limita a expressividade oferecida por NCL.

The 'Create Interactive Event' dialog box is used to define interactive events. It features three main sections: 'Remote Control', 'Triggering Interval', and 'Resulting Event'. The 'Remote Control' section includes a dropdown for '\* Key' (currently showing a blue sphere icon). The 'Triggering Interval' section includes dropdowns for '\* Relative to' (currently 'video1') and '\* Anchor' (currently 'lambda'), along with a 'New...' button. The 'Resulting Event' section includes dropdowns for '\* On' (currently 'ads1') and '\* Action' (currently 'start'), along with a 'New...' button. At the bottom, there are 'OK', 'Cancel', and 'Help' buttons, and a note '\* Required fields.'.

Figura 3.5: Definição dos elos de interatividade do Composer, retirado de [45]

Também é possível fazer a simulação de eventos interativos e assim visualizar a cadeia temporal atribuindo um valor para os tempos indeterminados devido à interatividade. A Figura 3.6 demonstra uma cadeia temporal caso o telespectador interaja com a aplicação em um determinado instante durante a apresentação do objeto *node0*, o que finaliza a apresentação do *node1*.

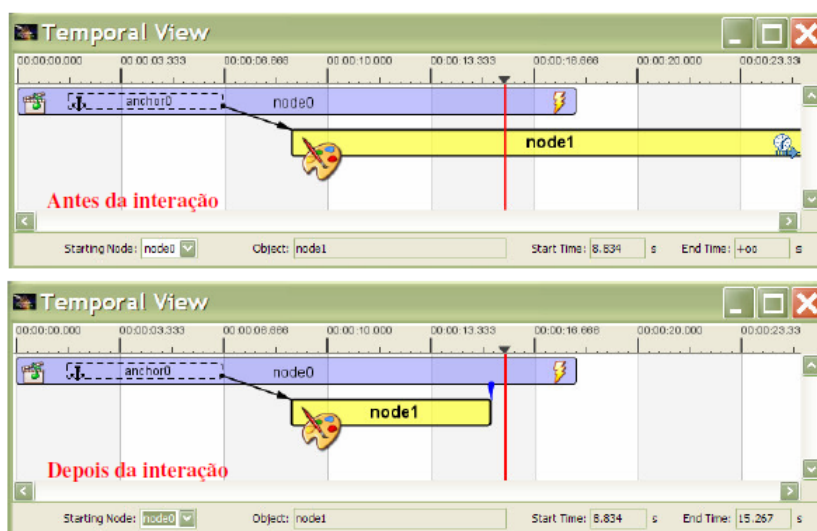


Figura 3.6: Simulação de eventos interativos do Composer, retirado de [45]

Apesar desta visão temporal proposta pela versão inicial do *Composer* apresentar funcionalidades que facilitam a autoria de documentos NCL, ela não está disponível na nova versão da ferramenta, *Composer 3* [53], que não oferece nenhuma visão temporal.

### 3.1.2 NEXT

NEXT [55] é uma ferramenta de autoria gráfica de documentos NCL com suporte a templates de composição, estruturas genéricas responsáveis por especificar os relacionamentos entre os componentes do documento. Tais templates são especificados na linguagem XTemplate 3.0 [59]. Assim, a ferramenta facilita e agiliza o desenvolvimento de aplicações por autores com pouco ou nenhum conhecimento em NCL para o sistema brasileiro de televisão digital interativa. NEXT oferece diferentes visões do documento que são disponibilizadas através de *plugins*, que são adicionados ao NEXT por meio de uma arquitetura extensível e adaptável a diferentes perfis de autores.

A arquitetura do NEXT é baseada em um núcleo apto a tratar documentos NCL e a comunicação com os *plugins*. Sendo assim, ela permite que novas funcionalidades sejam adicionadas ao editor facilmente. Na Figura 3.7, é possível notar que o núcleo é completamente independente dos *plugins*.

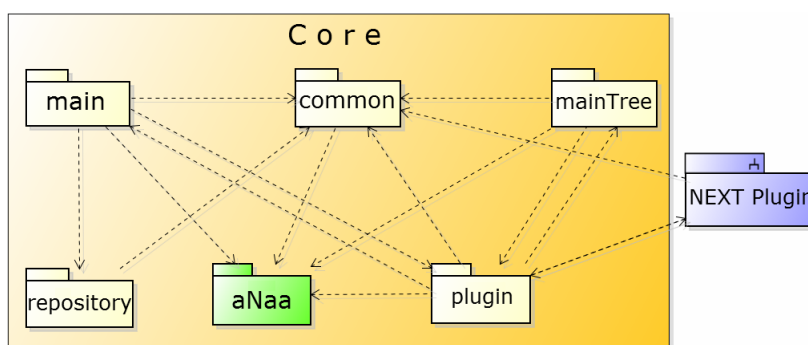


Figura 3.7: Arquitetura do NEXT

O pacote *aNaa* [42] é responsável pela importação, criação e edição de documentos NCL utilizando a API aNa [60]. Além disso, ela é capaz de analisar a consistência estrutural e comportamental do documento. Os pacotes *main* e *common* são responsáveis pela interface gráfica e pelo idioma da ferramenta respectivamente. O pacote *plugin* encarrega-se da troca de informações entre os *plugins* e o núcleo do NEXT, além de guardar informações sobre os mesmos e informá-los também sobre alterações no documento NCL. E o pacote *repository* armazena as mídias que são importadas ao NEXT para serem utilizadas na construção da aplicação.

Para adicionar uma nova funcionalidade ao NEXT através de criação de um *plugin*, deve-se desenvolver um programa em java com a extensão *jar*. Além disso, para prover a comunicação do *plugin* com o NEXT, deve-se desenvolver um pacote chamado de *myPlugin* contendo as classes *StartPlugin* e *PluginInfo*. A primeira deve estender uma classe do pacote *javax.swing*, *JInternalFrame*, para inicializar o *plugin*, o qual é chamado pelo núcleo do NEXT. Ela deve implementar os métodos *getUpdateAtNCLDocument* para notificar os *plugins* sobre modificações no documento NCL. A classe *PluginInfo* armazena informações dos *plugins* que deve conter dois métodos: *getAlias*, que corresponde ao nome do *plugin* no NEXT e *getNCLInterest*, que retorna uma lista de elementos de interesse do *plugin* que, quando forem modificados, o *plugin* é notificado.

O NEXT possui quatro *plugins*, sendo que um deles é o próprio editor gráfico de criação de documentos NCL com uso de templates de composição. Os templates são especificados através da linguagem XTemplate 3.0 [59]. Esse *plugin* utiliza um template como entrada e através da especificação do mesmo suas telas são geradas, indicando graficamente como os nós de mídia da aplicação serão apresentados simultaneamente na TV. O NEXT possui uma base de templates que podem ser escolhidos pelo autor para construir sua aplicação. Após a escolha do template, ele só necessita preencher os componentes das telas do template utilizando o repositórios de mídias. Os relacionamentos

entre os componentes do documento já são especificados pelo template. A Figura 3.8 mostra como o repositório de mídias é usado com o editor para preencher o template através do recurso *drag-and-drop*.

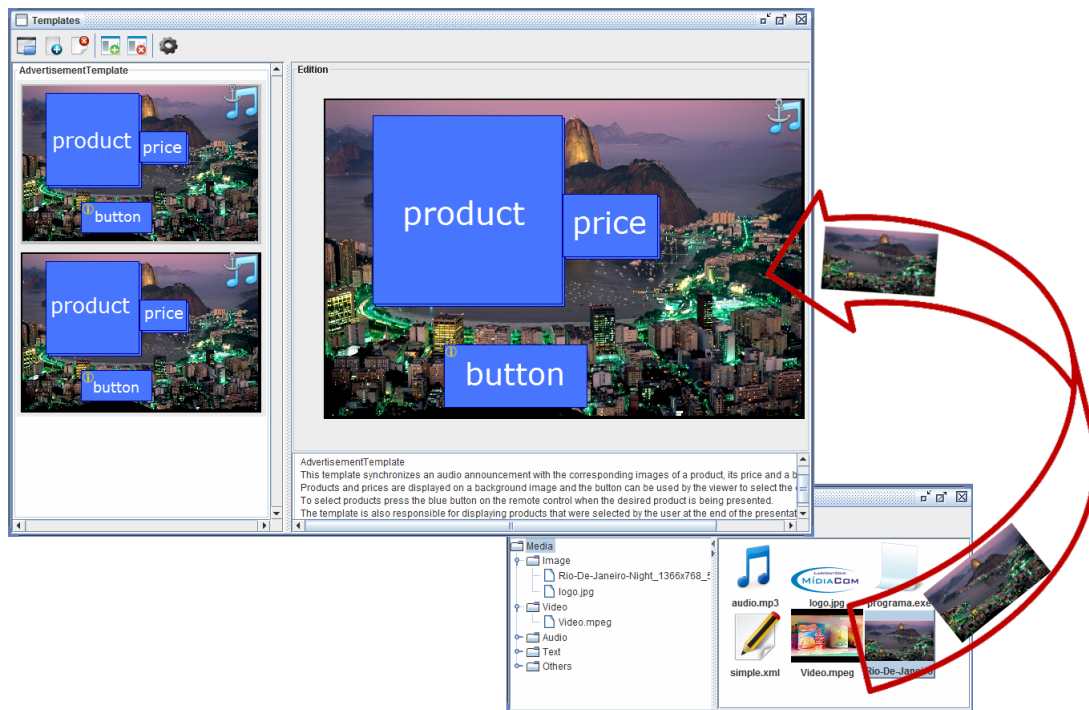


Figura 3.8: *Drag-and-drop* no NEXT

O *plugin* de visão estrutural permite a edição gráfica do documento NCL. Ainda que facilite o desenvolvimento de aplicações para TV digital, este *plugin* exige que o autor tenha pelo menos um conhecimento básico da linguagem NCL. A interface gráfica pode ser vista na Figura 3.9.

O *plugin* de visão de leiaute possibilita a criação e edição das regiões e descritores do documento NCL. Em sua interface, exibida na Figura 3.10, as regiões são construídas graficamente e redimensionadas apenas usando o mouse. Quanto aos descritores, estes devem ser editados através de caixas de diálogo que exigem do autor conhecimento básico em NCL. Esta ferramenta pode ser usada independentemente do NEXT, criando bases de regiões e descritores que poderão ser usadas por qualquer documento NCL.

O *plugin* de conectores permite criar e editar conectores hipermídia graficamente para documentos NCL. Como o *plugin* de visão de leiaute, ele permite ser usado separadamente do NEXT para criar uma base de conectores que pode ser utilizada por qualquer documento NCL. Sua interface gráfica é exibida na Figura 3.11.



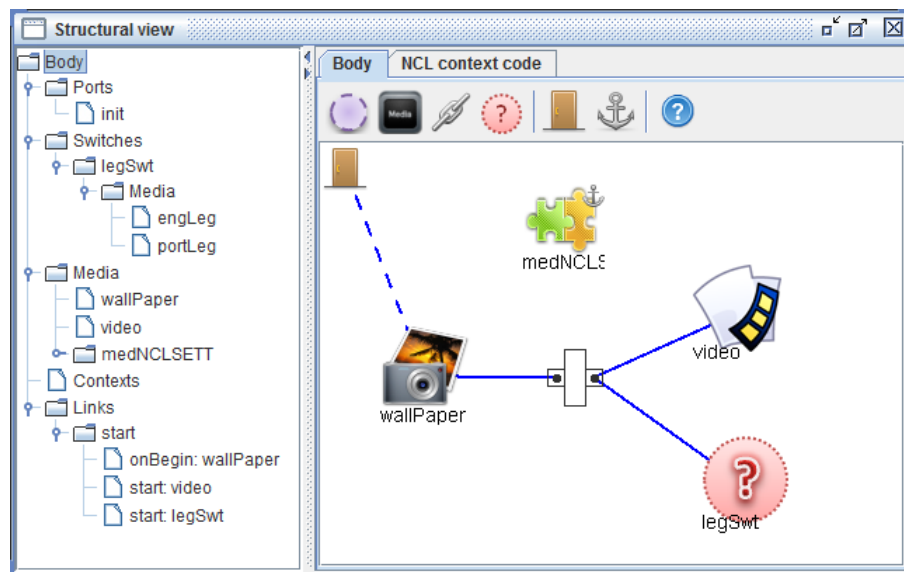


Figura 3.9: Visão Estrutural do NEXT

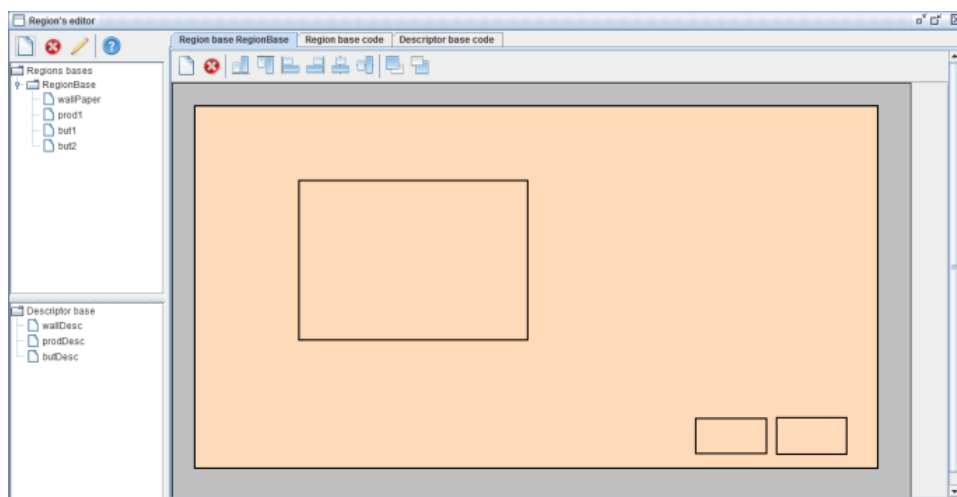


Figura 3.10: Visão de Leiaute do NEXT

### 3.1.3 CMIFed

O editor CMIFed [68] é um ambiente de edição e apresentação de documentos hipermídia. Ele utiliza um modelo de dados, o *CMIF hypermedia model*, que se baseia em outro modelo hipermídia, o *Amsterdam hypermedia model* [49], que por sua vez estende o *Dexter hypertext reference model* [47]. Este modelo permite a criação de apresentações aninhadas, onde a estrutura de aninhamento representa o sincronismo temporal do documento. Um documento CMIF pode ser representado por uma estrutura em árvore, cujos nós folhas são apresentações atômicas, e os galhos possuem semântica temporal sequencial ou paralela.

No modelo conceitual do CMIF, uma apresentação atômica é uma coleção de eventos. Estes representam fragmentos de áudio, vídeo, imagem ou texto. Marcadores são

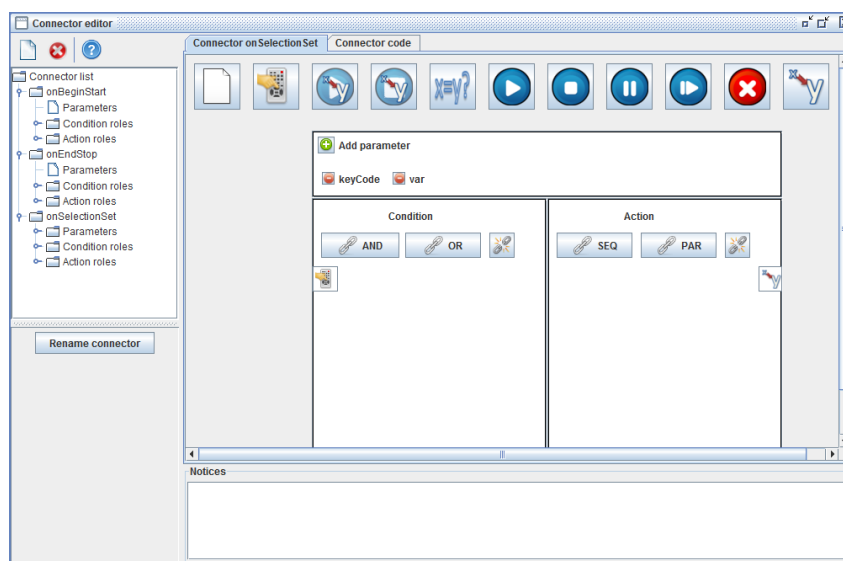


Figura 3.11: Editor de conectores do NEXT

âncoras que servem para relacionar, temporalmente, componentes do documento hiper-mídia. Propriedades específicas de apresentação são armazenadas com os eventos, como os marcadores também são. Existem três níveis de hierarquia: os descritores de eventos que especificam como eles ocorrem na apresentação, os descritores de dados que detalham propriedades estáticas dos dados e o dado propriamente dito.

Os eventos de uma apresentação atômica são ordenados através de relacionamentos temporais. Nesse modelo são usados dois mecanismos para expressar relações de sincronismo temporal: composição paralela e sequencial, e arcos de sincronização (sinc-arcs). A hierarquia da árvore do documento CMIF já define os relacionamentos temporais através da composição paralela e sequencial, ou seja, o paradigma de sincronização temporal é hierárquico no CMIF. Quanto aos arcos de sincronização, eles fazem o relacionamento entre os marcadores de eventos de uma mesma apresentação atômica especificando um retardo. A interatividade nas apresentações é representada através dos elos hiper-mídia (*hyperlinks*) que, como os sinc-arcs, ligam marcadores de eventos. Além disto, o modelo CMIF possui o conceito de *canal*, que agrupa os eventos de uma apresentação pelo tipo de mídia. Uma lista de atributos é definida para cada elemento do modelo detalhando suas propriedades.

CMIFed possui duas visões da apresentação para a edição do documento. Uma delas é a visão de hierarquia onde os eventos são aninhados formando subapresentações. Na Figura 3.12, pode ser notado que a apresentação *time-mgmt* possui 3 apresentações compostas (*intros*, *video scene* e *summary*). A ordem como elas são dispostas na janela (de cima pra baixo) define a relação temporal por composição sequencial, ou seja, a primeira

a ser exibida é a *intros*, em seguida *video scene* e *summary*. A primeira apresentação contém as apresentações atômicas (*title*, *still*, *intro* e *music*) dispostas uma ao lado da outra. Isto representa uma composição paralela indicando que tais apresentações são exibidas ao mesmo tempo.

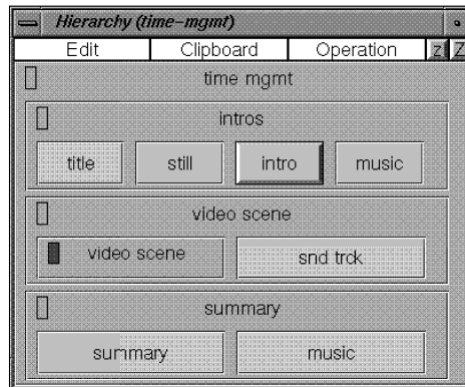


Figura 3.12: Visão hierárquica do CMIFed, retirado de [68]

A outra visão é a de canal onde os eventos são agrupados pelo tipo de mídia. A composição sequencial é representada pelo alinhamento na horizontal dos componentes e os mesmos dispostos na vertical dentro de um canal indicam a composição paralela. Esta visão é mostrada na Figura 3.13 onde, também, nota-se um arco de sincronização entre os componente *still* do canal *image* e *title* do *title text*.

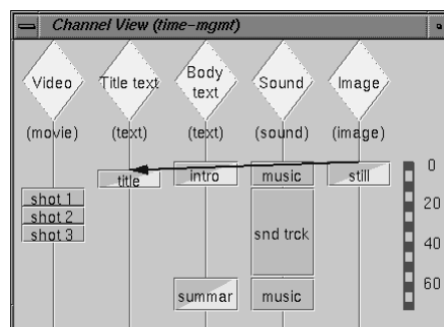


Figura 3.13: Visão de canal do CMIFed, retirado de [68]

O editor ainda possui um *player* onde é possível visualizar a aplicação desenvolvida. Ele possui uma janela de controle da apresentação e o leiaute das janelas pode ser alterado (geometria das janelas dos canais e o posicionamento das âncoras nas imagens). Esta visão é demonstrada na Figura 3.14.

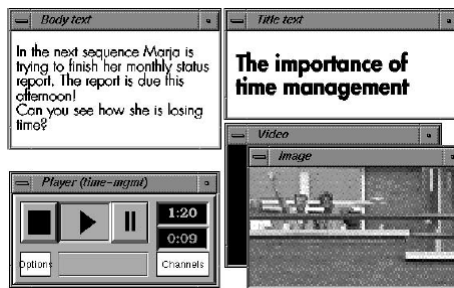


Figura 3.14: Player do CMIFed, retirado de [68]

### 3.1.4 FireFly

FireFly [33] é um sistema que facilita a criação, edição e apresentação de documentos hipermídia, ou seja, é uma ferramenta de autoria para o desenvolvimento de aplicações hipermídia. Ela utiliza um modelo de dados que divide um documento em três partes: itens de mídias, relacionamentos temporais e listas de operações. O sistema ainda possui um formatador que processa os relacionamentos temporais para controlar a apresentação do documento.

Os itens de mídias são fragmentos de objetos de mídia do documento que podem ser de vários tipos (vídeo, áudio, imagem, texto etc.). Tais itens especificam seu comportamento temporal na apresentação. Para isso, estes itens de mídia possuem eventos que indicam pontos, os quais podem ser relacionados com outros. Os eventos podem ser síncronos, o instante de tempo na apresentação do item pode ser previsto, ou assíncronos, quando o posicionamento no tempo é indeterminado. Cada item também tem associado um procedimento de controle que permite alterar o modo como as mídias se apresentam. As listas de operações realizam o mesmo que os procedimentos de controle, porém elas não são armazenadas juntamente com os itens de mídia. Desta forma, o modelo permite que as configurações de apresentação de uma lista de operações possam ser reutilizadas por outro item de mídia.

A sincronização temporal é realizada através de dois tipos de relacionamento temporal: de igualdade e desigualdade temporais. O primeiro indica que dois eventos são apresentados simultaneamente no tempo ou sequencialmente com intervalo de tempo fixo especificado entre eles. O relacionamento de desigualdade liga dois eventos que ocorrem sequencialmente, porém com intervalo de tempo, entre eles, indeterminado.

O FireFly apresenta duas ferramentas para visualização, criação e edição dos relacionamentos temporais. Uma delas é o visualizador de itens de mídia mostrado na Figura 3.15.

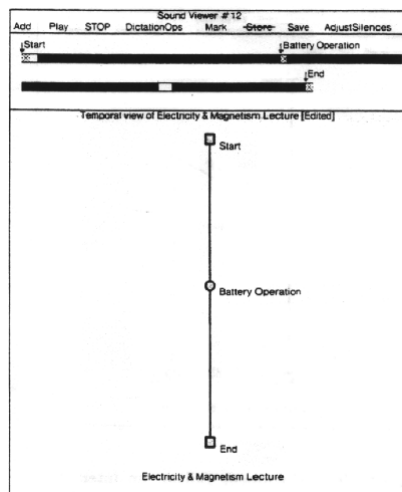


Figura 3.15: Visualizador de itens de mídia do FireFly, retirado de [33]

Na Figura 3.15, é representada, através de um grafo, a mídia de áudio, *Electricity & Magnetism Lecture* que possui dois retângulos, *Start* e *End* especificando os eventos de início e fim respectivamente. Além disso, um círculo indica um evento interno. As arestas que ligam tais eventos possuem um tamanho que é proporcional ao intervalo entre esses eventos.

A outra ferramenta, exibida na Figura 3.16, permite editar o documento hipermídia através de uma barra de menu. Note que o evento interno *Battery Operation* é usado para sincronizar com o evento de início do item de mídia *Battery Animation*.

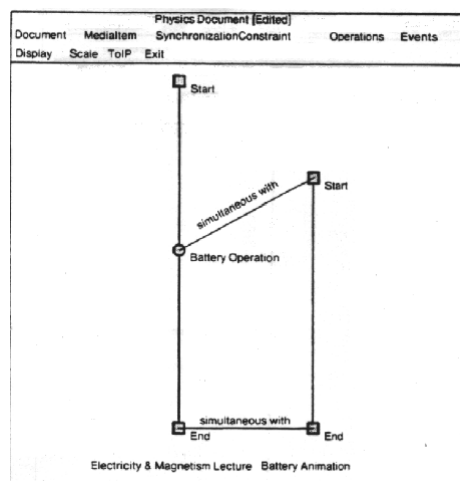


Figura 3.16: Editor interativo de documentos do FireFly, retirado de [33]

### 3.1.5 GRiNs

GRiNs [35] é uma ferramenta de autoria de documentos hipermídia, baseados na linguagem SMIL 2.0 [7], com ênfase no desenvolvimento das aplicações através da visão temporal utilizando recursos gráficos. O documento criado pode ser visualizado na própria ferramenta. A visão temporal cria os documentos de acordo com os elementos de sincronização temporal especificados na linguagem de autoria.

Os elementos da linguagem SMIL que permitem definir a ordem temporal das mídias na apresentação são: sequencial, paralelo e exclusivo. Cada um pode conter vários objetos de mídias indicando, no caso do elemento sequencial, que todos são apresentados em sequência na execução do documento. É possível fazer o aninhamento destes elementos, ou seja, um contêiner sequencial pode possuir um paralelo e vice-versa, assim criando a sincronização entre as mídias conforme desejado.

A visão hierárquica do GRiNs, como exibida na Figura 3.17, agrupa os objetos de mídia de acordo com sua semântica temporal, isto é, os objetos que se apresentam em sequência são colocados em retângulos que indicam uma ordem sequencial e analogamente para os objetos que são exibidos em paralelo ou que devem ser escolhidos (contêiner *switch*).

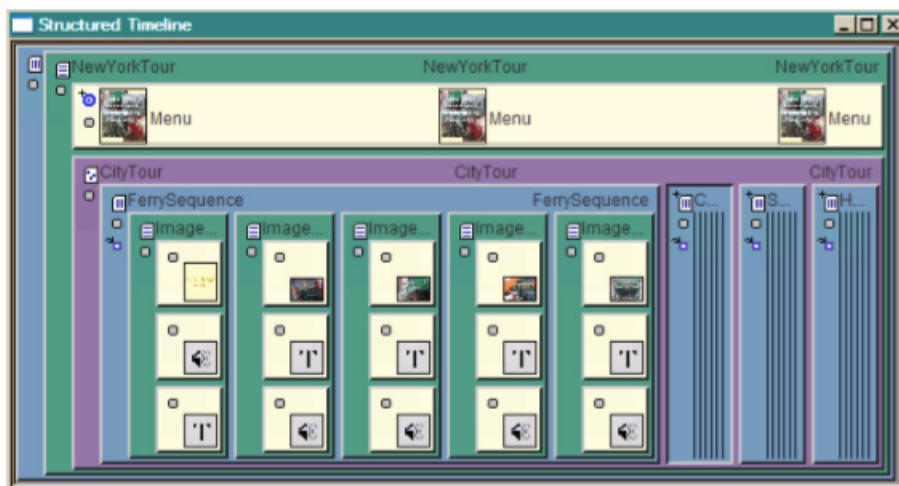


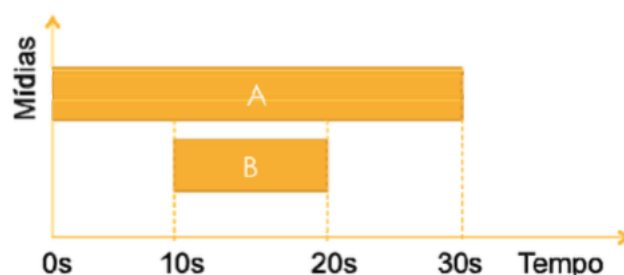
Figura 3.17: Visão temporal do GRiNs , retirada de [36]

Na Figura 3.17, o documento *Sample-Slideshow* possui o contêiner *Audio-and-Images* que representa um grupo paralelo, o *Image-Sequence* que indica um conjunto de mídias, as quais são apresentadas em sequência e o *Audio-Switch* que simboliza uma escolha, no caso, a opção entre o áudio em espanhol ou inglês.

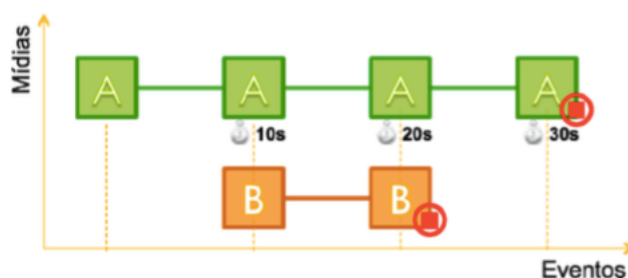
### 3.1.6 Eventline

Eventline [37] é uma proposta de representação da visão temporal de aplicações multimídia fundamentada no paradigma baseado em eventos. Nesta representação, o eixo temporal é discretizado em eventos, definidos pelo modelo NCM, que ocorrem sobre uma mídia em um determinado instante de tempo. Estas ocorrências de eventos são dispostas no eixo de maneira relativa aos eventos que são disparados por uma mídia denominada de principal. No caso de interatividade, múltiplas *eventlines* (linhas de eventos) são necessárias, as quais possuem a mesma mídia principal. Este trabalho não realiza a implementação da abstração proposta, ele a sugere como uma representação a ser seguida pelas ferramentas que oferecem a visão temporal de aplicações multimídia.

A Figura 3.18 ilustra uma aplicação na *timeline* convencional e a mesma na *eventline*. A Figura 3.18a mostra uma mídia A que inicia a aplicação e termina no instante de tempo 30s. Após 10s do seu início, uma nova mídia B é iniciada. No instante 20s, a mídia B termina sua apresentação. Discretizando esse eixo de tempo, conforme mostrado na Figura 3.18b, é considerado que o início de A representa um evento de ativação e que a mídia B também passa por um evento de ativação quando A passa por um evento temporal de 10s. Quando A passa por um evento temporal de 20s, B passa por um evento de término. E a mídia A passa por um evento de fim quando ela passa por um evento temporal de 30s.



(a) Aplicação na *timeline* convencional



(b) Aplicação na *eventline*

Figura 3.18: Diferenças entre *timeline* e *eventline*, retirada de [37]

### 3.1.7 LimSee 2

A ferramenta LimSee 2 [41] oferece uma visão temporal para editar a estrutura temporal da linguagem SMIL. Assim, os objetos de mídia são representados graficamente por retângulos cujo comprimento está diretamente relacionado à duração do objeto e sua posição no eixo do tempo representa seus instantes de início e término. LimSee 2 possui diversas visões do documento: temporal, espacial e textual. A visão temporal do LimSee 2 pode ser visualizada na Figura 3.19.

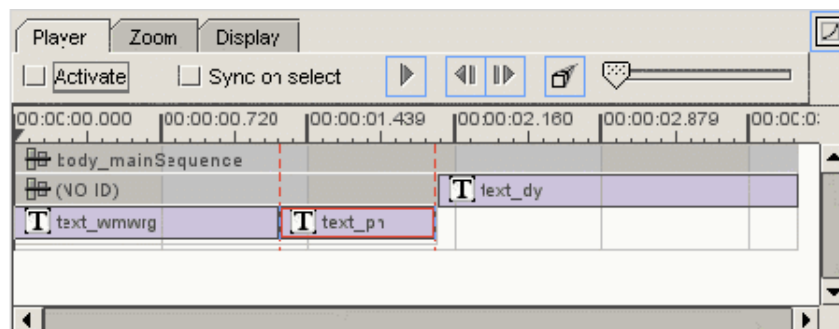


Figura 3.19: Visão temporal do LimSee 2 , retirada de [22]

A ferramenta contém diversas das funcionalidades relacionadas a restrições contidas na ferramenta Madeus [67], a qual deu origem ao LimSee 2, porém adicionando suporte especificação de leiaute e restrições de tempo. LimSee 2 suporta também agrupamento de objetos em composições de sincronização abstratas, o que caracteriza a ferramenta como utilizadora de outra abordagem de autoria, a de sincronização hierárquica. Entretanto, a ferramenta não suporta composições atemporais. Não existe uma relação visual entre a *timeline* e as restrições de leiaute. Além disso, eventos assíncronos e a adaptação de conteúdo não são suportados pelo sistema.

## 3.2 Ferramentas Comerciais

O desenvolvimento de apresentações multimídia em softwares comerciais é comumente realizado com o paradigma de linha do tempo (*timeline*), ou seja, os objetos de mídia são posicionados diretamente no eixo temporal e, intuitivamente, o autor cria seu documento multimídia colocando os componentes da apresentação nos instantes em que eles devem ser exibidos. Como exemplos destas ferramentas, têm-se o *Final Cut Pro* [20] e *iMovie* [21] da *Apple*, que fazem edição profissional de vídeos e editam filmes caseiros respectivamente. As interfaces dessas ferramentas são mostradas na Figura 3.20.





(a) Final Cut Pro

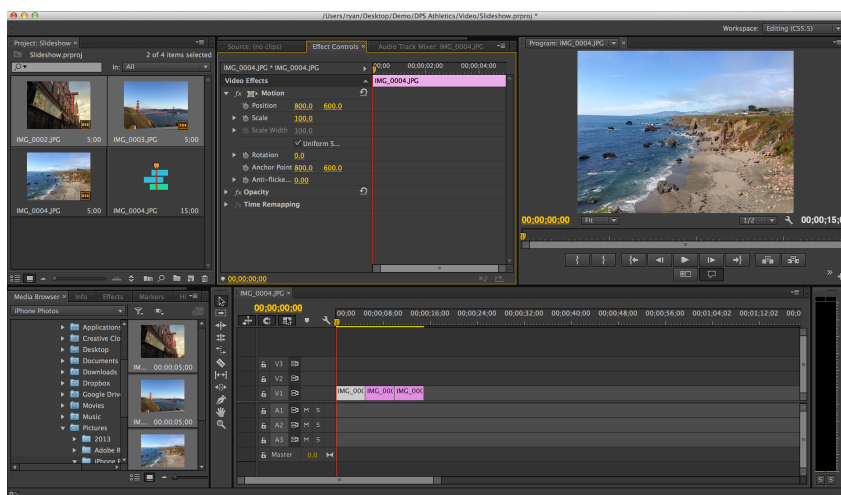


(b) iMovie

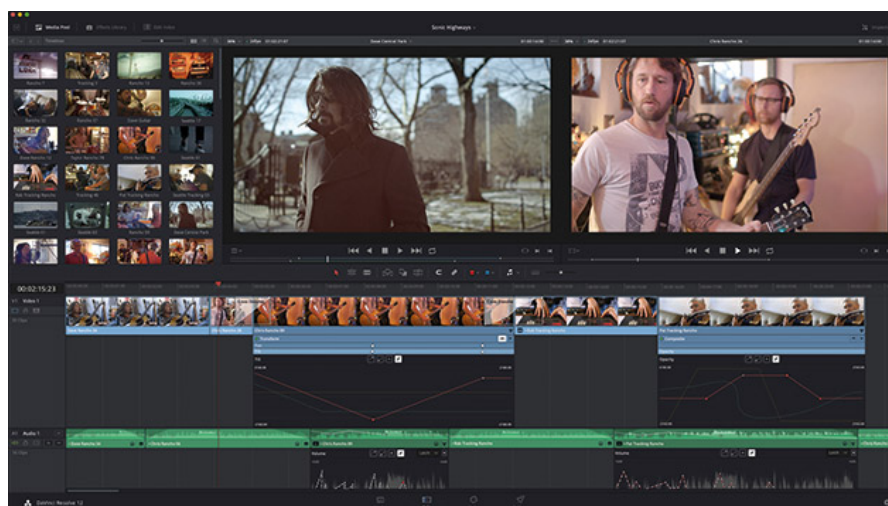
Figura 3.20: Ferramentas de edição de conteúdo multimídia, Apple

Além destas, há outro editor de vídeos profissional o *Adobe Premiere* [18] da *Adobe* que juntamente com outras ferramentas da *Adobe* oferece recursos similares ao *Final Cut Pro*. Uma alternativa gratuita a estas ferramentas, lançado em junho de 2015, é o editor *Davinci Resolve 12* [19], antes um poderoso software de correção de cor, que passou a realizar edição de vídeos por meio da *timeline* integrada a diversos recursos. As interfaces gráficas dessas ferramentas são exibidas nas Figuras 3.21a e 3.21b respectivamente.

Outra ferramenta de edição de vídeo que também utiliza o paradigma de *timeline*, porém mais simples, é o sistema *Nero Video 2016* [25] da *Nero AG* que separa, explicitamente, as mídias da linha do tempo em canais como mostrado na Figura 3.22.



(a) Adobe Premiere



(b) Davinci Resolve 12

Figura 3.21: Ferramentas profissionais de edição de vídeo



Figura 3.22: Nero Video 2016, Nero AG

### 3.3 Requisitos de um Editor Gráfico para Autoria de Documentos Hiperímia

A partir do estudo realizado dos trabalhos relacionados, tanto acadêmicos quanto comerciais, foram selecionados requisitos relevantes para o desenvolvimento de uma ferramenta de autoria multimímia. Ela deve ser capaz de gerar documentos especificados em linguagens de autoria multimímia padrão, suportar a definição de interatividade e de relações temporais para a sincronização temporal, fazendo uso de uma interface gráfica simples que integre a visão temporal e espacial permitindo que autores sem conhecimento de linguagem ou modelo multimímia possam criar suas próprias aplicações hiperímia e executá-las em diferentes plataformas. Estes requisitos estão listados a seguir:

- Formato final da aplicação: o usuário deve ser capaz de exportar as aplicações hiperímia criadas na ferramenta para documentos especificados em alguma linguagem padrão de autoria multimímia, para que a aplicação possa ser executada em diferentes plataformas;
- Edição da visão temporal: a ferramenta deve oferecer a visão temporal das aplicações hiperímia, onde os itens de míia são dispostos diretamente no eixo do tempo. Além disso, ela deve permitir que o usuário edite a aplicação através desta visão utilizando relações temporais;
- Edição da visão espacial: a ferramenta deve oferecer a visão espacial das aplicações hiperímia mostrando como e onde os itens de míia são apresentados no momento da execução das aplicações. Além de permitir que o usuário realize a edição das características de apresentação das míias (ex.: posição e tamanho);
- Suporte à interatividade: o autor deve ser capaz de definir relações de interatividade, onde a interação do usuário com algum componente da aplicação altera sua apresentação;
- Prévia da execução: a ferramenta deve permitir que o usuário visualize o comportamento temporal e espacial de suas aplicações hiperímia quando elas forem executadas. Esta previa da aplicacao nao permite que o autor realize interações com os componentes da aplicação, uma vez que ela e uma simulacao e nao a execucao da aplicacao propriamente dita;
- Suporte a autores sem conhecimento de modelo ou linguagem: a ferramenta deve permitir que autores sem conhecimento de modelos hiperímia (ex.: NCM) ou de

linguagens de autoria multimídia (ex.: NCL, SMIL e HTML5) consigam criar aplicações multimídia interativas para diversas plataformas;

- Execução: a execução da aplicação hipermídia deve ser oferecida dentro do próprio ambiente de autoria da ferramenta.

## 3.4 Comparação dos Trabalhos Relacionados

Dados os requisitos levantados na Seção 3.3 para o desenvolvimento de um editor gráfico para autoria de documentos hipermídia, a Tabela 3.1 resume a disponibilidade desses recursos em cada uma das ferramentas acadêmicas de autoria multimídia estudadas neste capítulo.

É importante ressaltar que o trabalho *Eventline* não está presente na Tabela 7.1, visto que ele propõe apenas um modelo de representação da visão temporal de aplicações multimídia, utilizando um eixo discreto baseado em eventos de forma a vencer as limitações do paradigma *timeline*, e não implementa a abstração proposta oferecendo uma ferramenta de autoria multimídia.

Em relação às ferramentas comerciais citadas neste capítulo, todas utilizam o paradigma de autoria *timeline*. Além disso, elas oferecem a edição da visão temporal e espacial, e a execução da aplicação. A análise do requisito de prévia da execução para estas ferramentas comerciais não é aplicável, visto que elas não suportam a interatividade. Apesar destas ferramentas comerciais permitirem que usuários sem conhecimento de modelos e linguagens de autoria multimídia possam criar suas aplicações, elas geram um arquivo de vídeo (ex.: MP4 e MOV) como formato final da aplicação hipermídia, não permitindo exportar para alguma linguagem padrão de autoria multimídia.

## 3.5 Conclusão

Este capítulo discutiu diversos trabalhos que apresentam ferramentas que auxiliam na autoria de documentos hipermídia, enfatizando como a visão temporal destes documentos é tratada por estes trabalhos. Dentre elas, a ferramenta *CMIFed* apresenta uma visão temporal estruturada hierarquicamente através de composições paralelas e sequenciais como a ferramenta *GRiNs*. E o sistema *FireFly* utiliza grafos para representar a visão temporal. Além destas ferramentas, foram analisados alguns softwares comerciais que

Tabela 3.1: Recursos das Ferramentas de Autoria

Recurso / Editor	Composer	NEXT	CMIFed	FireFly	GRiNs	LimSee 2
Paradigma de autoria	E	E	H	R	H	H
Formato final da aplicação	NCL	NCL	SMIL	Modelo FireFly	SMIL	SMIL
Edição da Visão temporal			✓		✓	✓
Edição da Visão espacial	✓	✓	✓		✓	✓
Suporte à interatividade	✓	✓	✓	✓	✓	
Prévia da execução			✓	✓	✓	✓
Suporte a autores sem conhecimento de modelo ou linguagem		✓				
Execução			✓	✓	✓	✓

E : Baseado em Eventos; R : Baseado em Restrições; H : Hierárquico;

✓ : contemplado pelo editor

permitem a criação de apresentações multimídia e utilizam o paradigma de sincronização temporal baseado em *timeline*.

Dentre as ferramentas estudadas, apenas o *Composer* e o NEXT auxiliam na autoria de documentos NCL. O *Composer* apresenta diversas visões de um documento NCL, inclusive a visão temporal que utiliza um modelo de grafo para representar a cadeia temporal, e trata a interatividade em documentos hipermídia. Entretanto, em sua nova versão, nenhuma visão temporal está disponível. Além desta ferramenta para documentos NCL, foi apresentado o NEXT, que é um editor gráfico com suporte a templates de composição. Este editor também possui diversas visões do documento NCL, porém não oferece a visão temporal.

A ferramenta proposta nesta dissertação implementa a visão temporal dos documentos hipermídia utilizando o paradigma de autoria baseado em eventos, onde o autor pode especificar relação temporais causais entre os itens de mídia. Além disso, as aplicações criadas na ferramenta são facilmente exportadas para linguagens padrões de autoria multimídia como o NCL e HTML5.

# Capítulo 4

## SIMM - Simple Interactive Multimedia Model

Este capítulo descreve o modelo de documentos hipermídia proposto, chamado SIMM (*Simple Interactive Multimedia Model*). Primeiramente, o modelo proposto é introduzido na Seção 4.1. A hierarquia de classes que representam as entidades do modelo proposto é analisada na Seção 4.2. A Seção 4.3 discute o paradigma de sincronização temporal utilizado pelo SIMM e também as relações causais síncronas oferecidas pelo modelo. O suporte a relações de interatividade dado pelo SIMM é estudado na Seção 4.4. As entidades do modelo proposto que representam as características de apresentação dos itens de mídias durante a execução do documento hipermídia é discutida na Seção 4.5. Em seguida, a sincronização espacial oferecida pelo SIMM é apresentada na Seção 4.6. O capítulo ainda discute as limitações do modelo SIMM em relação ao NCM na Seção 4.7.

### 4.1 Introdução ao SIMM

Esta seção introduz o modelo de documento hipermídia proposto neste trabalho, SIMM (*Simple Interactive Multimedia Model*). Ele visa facilitar a autoria de aplicações multimídia interativas utilizando o modelo de sincronização temporal baseado em eventos. Este se baseia na ocorrência de eventos durante a execução do documento para especificar as relações temporais condicionais entre os itens de mídia.

SIMM é proposto visando ser um modelo base para o desenvolvimento de ferramentas de autoria multimídia para usuários não especialistas, fornecendo um modelo simples, capaz de representar as classes mais relevantes de aplicações hipermídia e que seja independente de linguagens de autoria multimídia. Dessa forma, o modelo proposto não visa

substituir modelos hipermídia mais expressivos presentes na literatura, por exemplo, o modelo NCM.

O modelo proposto é baseado nas entidades básicas de um modelo hipermídia, nós e elos, e nos seus tipos de eventos: de apresentação e de seleção. O primeiro refere-se à apresentação de um item de mídia durante a execução do documento hipermídia, conforme suas características de apresentação definidas no modelo, as quais serão discutidas em detalhes na Seção 4.5. Os eventos de seleção referem-se à interação do espectador com a aplicação multimídia através da seleção de um item de mídia, o qual é dito como mídia interativa, utilizando um dispositivo de entrada como o mouse, teclado ou controle remoto. O suporte do modelo à interatividade será discutida na Subseção 4.4.

Os elos temporais do modelo proposto são definidos baseando-se nas relações temporais entre intervalos apresentadas em [27], para as quais, o modelo SIMM atribui uma interpretação de causalidade. Além disso, o modelo estende estas relações temporais permitindo especificar um valor de tempo como atraso no disparo da ação definida no elo temporal. As relações temporais no SIMM são compostas por uma condição e ações. O paradigma de sincronização temporal utilizado no modelo hipermídia proposto será abordado em detalhes na Seção 4.3. Além do modelo permitir especificar a sincronização temporal, ele define entidades para realizar a sincronização espacial, a qual será discutida na Seção 4.6.

## 4.2 Entidades

Na Figura 4.1, a sincronização espaço-temporal do documento hipermídia é especificada através de uma lista de cadeias temporais e de relações espaciais, as quais são representadas pelos atributos *temporalChainList* e *spatialRelationList* da classe *SpatialTemporalView* respectivamente. A hierarquia das classes que representam o modelo hipermídia proposto é baseada na definição de nós, elos e cadeias temporais, os quais são representados pelas classes *Media*, *Relation* e *TemporalChain* respectivamente.

Os nós são definidos como um trecho de informação e elos como relacionamentos entre estes nós. A classe *Media* possui como atributos básicos um nome (*name*), uma referência ao objeto do arquivo que corresponde ao conteúdo da mídia (*mediaFile*), o caminho do arquivo de mídia (*path*), o tipo de mídia que pode assumir os valores imagem, vídeo, áudio e texto (*mediaType*) e um objeto que representa as características de apresentação do item de mídia (*presentationProperty*), as quais serão detalhadas na Seção 4.5.

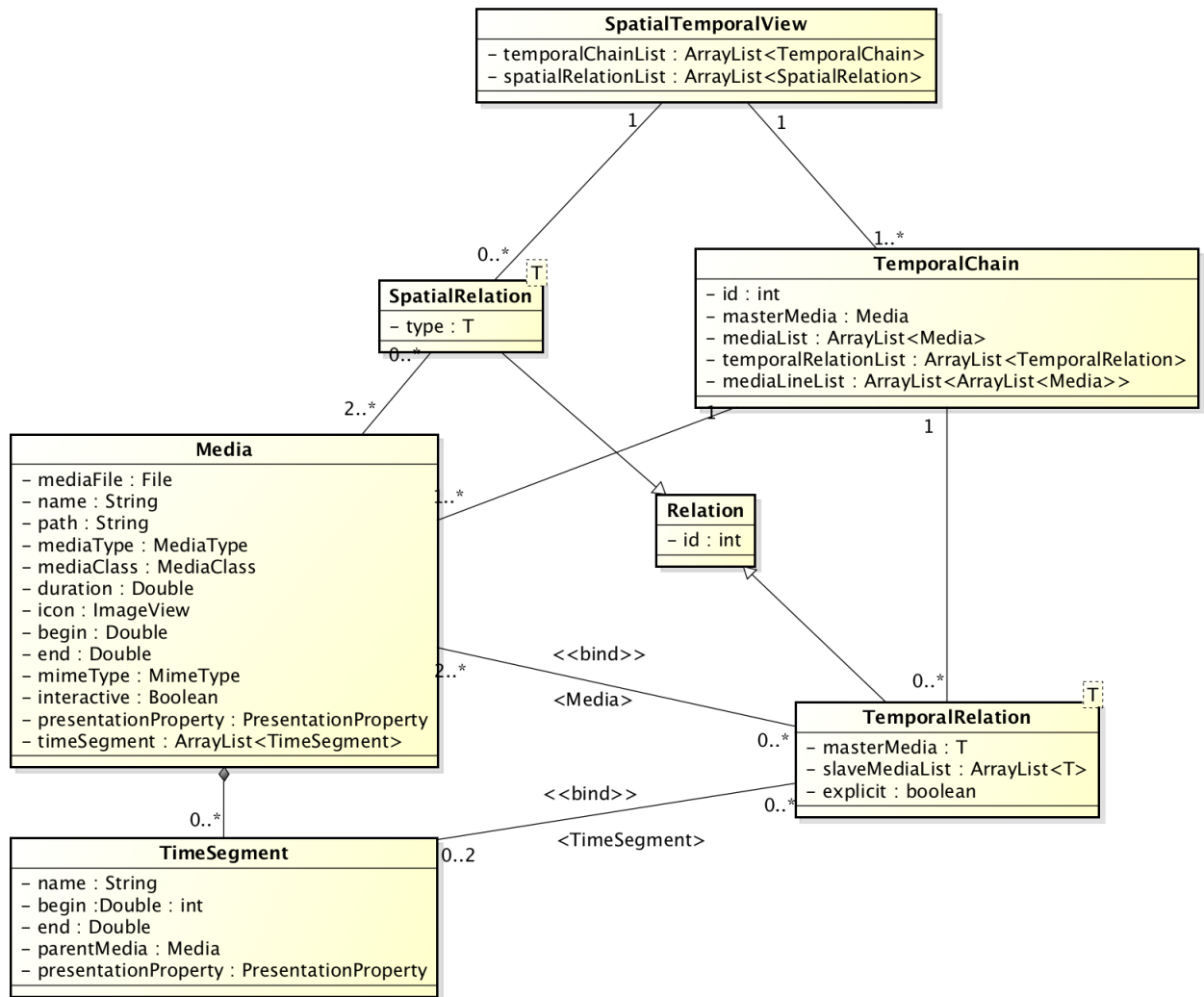


Figura 4.1: Diagrama de classes do modelo de sincronização espaço-temporal do SIMM

Para permitir a especificação de trechos de mídias contínuas, mídias podem ser compostas por entidades da classe *TimeSegment*. Esta define um trecho do conteúdo de mídias contínuas através da especificação dos instantes de início e fim com atributos *begin* e *end* do segmento de tempo. Além disso, cada trecho possui suas próprias características de apresentação.

A classe *Relation*, que contém apenas o atributo *id* (identificador da relação), possui duas especializações: *TemporalRelation* e *SpatialRelation*. A primeira possui como atributos uma mídia mestre (*masterMedia*) que representa a mídia dominante numa relação e uma lista de mídias denominadas escravas *slaveMediaList* da relação. Já a segunda apresenta apenas um atributo referente ao tipo do elo espacial, o qual será detalhado na Seção 4.6. A sincronização temporal será estudada em detalhes na Seção 4.3

Além de ser possível especificar um item de mídia para participar da relação temporal,



um trecho de mídia contínua, representado pela classe *TimeSegment*, também pode fazer parte de um elo temporal. Desta forma, a classe *TemporalRelation* é parametrizada generalizando o tipo de classe que pode ser associada ao atributo *masterMedia* e os tipos de elementos que podem ser adicionados à lista de escravos. No diagrama de classe 4.1, isto é representado pelo estereótipo *bind* e sua informação, que pode assumir os valores *Media* ou *TimeSegment*.

Por fim, a hierarquia das classes do modelo SIMM possui, como uma de suas entidades, as cadeias temporais. Uma cadeia temporal é uma sequência de ocorrências de eventos síncronos durante a apresentação de um documento como definido em [34]. Quando uma cadeia temporal inicia a apresentação de uma aplicação multimídia, esta é denominada como cadeia principal.

A Figura 4.2 exibe a visão temporal de um documento simples, onde um vídeo é apresentado inicialmente e depois de alguns instantes, uma imagem é exibida representando um botão para indicar que o espectador pode interagir com a mídia através de um dispositivo de entrada. Caso haja a interação, o documento inicia um áudio e termina a apresentação da imagem do botão. A visão temporal que representa este cenário possui duas cadeias temporais indicadas por Cadeia Temporal 1 e 2 na figura. A segunda cadeia é derivada do evento assíncrono que corresponde à interação do espectador com o botão.

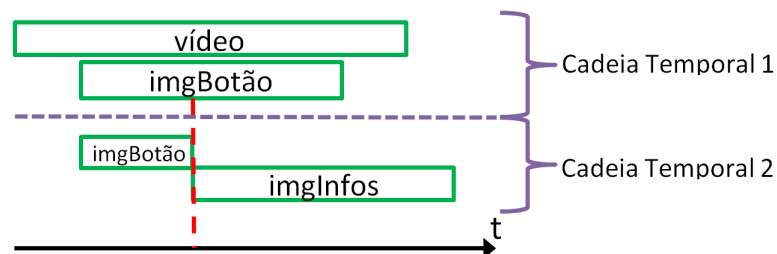


Figura 4.2: Visão temporal de um documento hipermídia

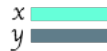
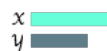

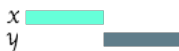

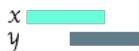

A classe *TemporalChain* que representa as cadeias temporais possui como atributos um identificador (*id*), uma mídia mestre (*masterMedia*) da cadeia temporal, ou seja, a mídia que inicia a cadeia e uma lista de mídias (*mediaList*) e de relações (*relationList*) que representam todas os itens de mídias contidos na cadeia temporal e todas as relações temporais entre os nós da cadeia respectivamente.

Nas seções seguintes, o modelo de sincronização temporal e espacial será detalhado apresentando os tipos de relação definidos e como a interatividade é suportada pelo modelo. Além disso, são comentadas as características de apresentação dos itens de mídia suportadas pelo modelo SIMM e suas limitações em relação ao NCM.

## 4.3 Sincronização Temporal

Utilizando o paradigma de sincronização temporal baseado em eventos, o modelo apresentado neste trabalho permite relacionar os itens de mídia de um documento no tempo através dos eventos que ocorrem durante a apresentação multimídia interativa. As relações temporais causais definidas no modelo são baseadas nas treze relações básicas entre intervalos de Allen [27] mostradas na Tabela 4.1.

Tabela 4.1: Relações Temporais entre Intervalos de Allen

Relação	Representação Gráfica
$x$ equals $y$	
$x$ starts $y$ $y$ started by $x$	
$x$ finishes $y$ $y$ finished by $x$	
$x$ meets $y$ $y$ met by $x$	
$x$ during $y$ $y$ contains $x$	
$x$ overlaps $y$ $y$ overlapped by $x$	
$x$ before $y$ $y$ after $x$	

A relação *equals* indica que o início e fim do intervalo  $x$  devem coincidir com o início e fim de  $y$  respectivamente. Já a relação *starts* representa que o início de  $x$  deve coincidir com o início de  $y$ . A relação *finishes* especifica que o término do intervalo  $x$  deve ocorrer ao mesmo tempo em que  $y$  finaliza. No caso da relação *meets*, o fim do intervalo  $x$  deve ocorrer no mesmo tempo em que o intervalo  $y$  inicia. Para a relação *during*, o intervalo  $x$  deve estar totalmente contido na duração de  $y$ . A relação *overlaps* define que o intervalo  $x$  deve sobrepor o intervalo  $y$  após um instante  $t$  de seu início. A última relação descrita na Tabela 4.1, *before* especifica que o intervalo  $x$  deve iniciar e terminar antes do início do intervalo  $y$ .

As relações causais do modelo permitem a definição de um instante de tempo representando o atraso no disparo das ações associadas à condição dos elos temporais. O modelo proposto suporta o início e fim do evento de apresentação e o evento de seleção. As relações propostas foram definidas de forma que o modelo possa representar os cenários temporais mais correntes no âmbito da produção de apresentações multimídia. A Figura 4.3 apresenta o diagrama de classes com as entidades relativas à representação da

sincronização temporal.

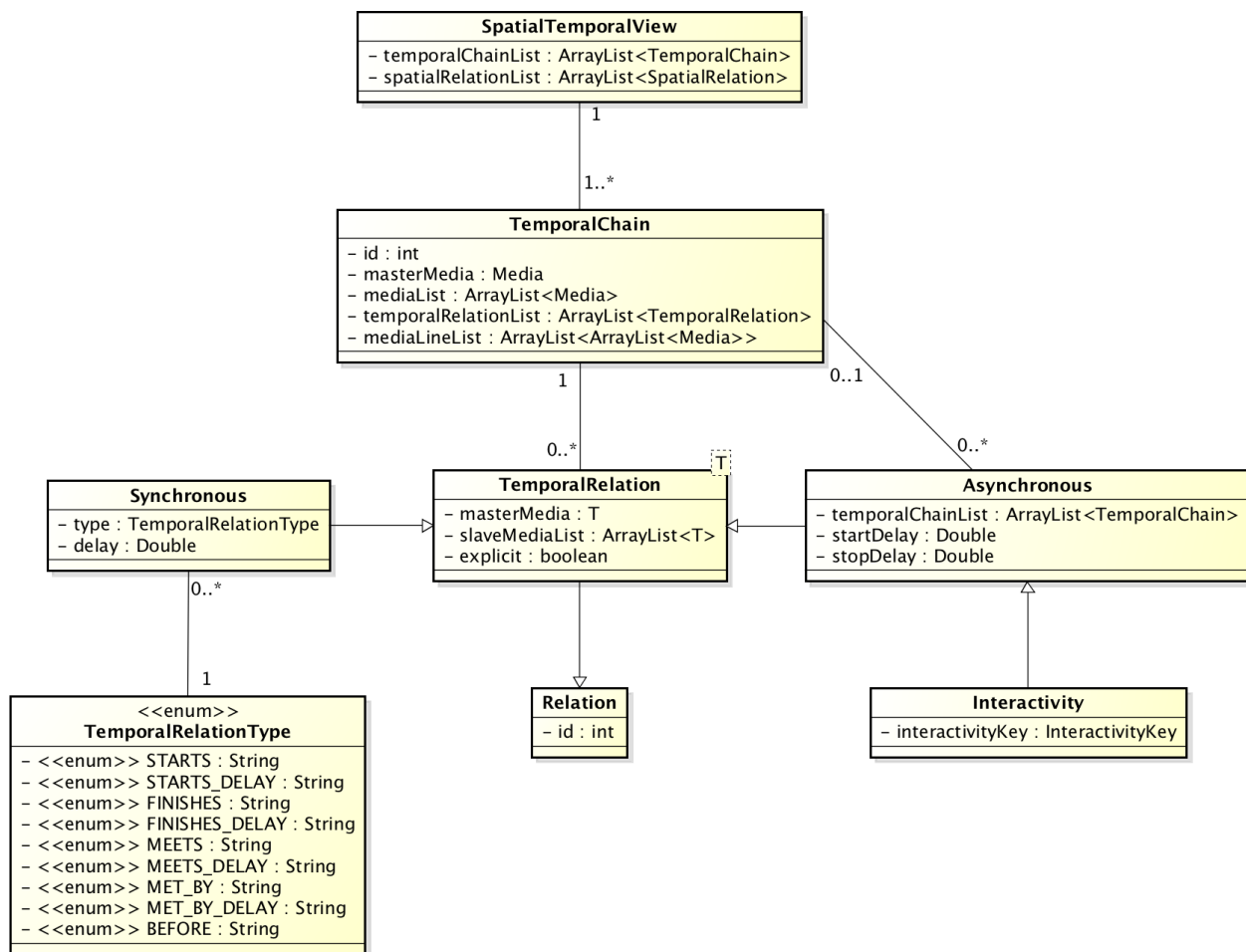


Figura 4.3: Modelo da sincronização temporal

No diagrama de classes apresentado na Figura 4.3, a visão temporal de documentos hipermídia é composta por uma ou mais cadeias temporais (*TemporalChain*). Cada cadeia temporal é composta por mídias e relacionamentos entre elas. As relações temporais no modelo proposto possuem uma mídia denominada como mestre representada pelo atributo *masterMedia* presente na classe *TemporalRelation*. As mídias mestres participam do elo temporal assumindo o papel da condição que deve ser satisfeita para que a ação associada ao mesmo elo seja disparada.

**Definição 1.** *Uma mídia mestre ou dominante, em uma relação temporal, é um nó cuja transição de estado de seu evento de apresentação (starts ou stops) ou de seleção (starts ou stops) dispara uma ação sobre outro nó alterando o estado do evento de apresentação deste.*

Além da mídia mestre, a relação temporal possui uma lista de mídias denominadas escravas representadas pelo atributo *slaveMediaList*. Estas mídias sofrem alteração em

seus estados de seus eventos de apresentação após a condição do elo, a qual pertencem, ser satisfeita. Essa alteração pode corresponder à ação de iniciar ou parar a apresentação da mídia escrava.

**Definição 2.** *Uma mídia escrava, em uma relação temporal, é um nó cuja transição de estado de seu evento de apresentação (starts ou stops) é disparada após a condição do elo temporal ao qual pertence ser satisfeita.*

Assim, o modelo proposto suporta somente as transições *starts* e *stops* para o evento de apresentação e de seleção entre as transições definidas no modelo NCM, conforme estudado no Capítulo 2. A Tabela 4.2 resume as transições e eventos considerados no tratamento das relações temporais causais no modelo proposto se fossem representadas por conectores NCM.

Nome do <i>role</i>	Nome da Transição	Tipo do evento
<i>onBegin</i>	<i>starts</i>	<i>presentation</i>
<i>onEnd</i>	<i>stops</i>	<i>presentation</i>
<i>onSelection</i>	<i>starts</i>	<i>selection</i>
<i>onSelection</i>	<i>stops</i>	<i>selection</i>

(a) Papéis de Condição

Nome do <i>role</i>	Ação	Tipo do evento
<i>start</i>	<i>start</i>	<i>presentation</i>
<i>stop</i>	<i>stop</i>	<i>presentation</i>

(b) Papéis de Ação










Tabela 4.2: Papéis suportados no modelo hipermídia proposto

A relação temporal definida no modelo possui duas classificações: síncrona e assíncrona. A classe *Synchronous*, que é uma especialização da classe *TemporalRelation*, representa as relações síncronas e possui o atributo *type* que pode assumir os seguintes valores, conforme a classe *TemporalRelationType* enumera: *Starts*, *Starts\_Delay*, *Finishes*, *Finishes\_Delay*, *Meets*, *Meets\_Delay*, *Met\_By*, *Met\_By\_Delay* e *Before*. Dependendo do tipo da relação de sincronização temporal, um valor de atraso  $t$ , representado pelo atributo *delay*, pode ser especificado indicando que a ação associada à relação deverá ser executada após  $t$  instantes de tempo da condição da relação ser satisfeita.

Estas relações possuem condições simples, ou seja, apenas uma condição precisa ser satisfeita para disparar a ação. Elas também definem apenas um tipo de ação a ser executada sobre um ou mais nós (mídias escravas). Por exemplo, suponhamos três mídias  $a$ ,  $b$  e  $c$ , e a relação  $r$  do tipo *Start\_Delay*. Definindo a mídia  $a$  como mestre de  $r$  e as

outras como escravas, esta relação iniciará  $b$  e  $c$   $t$  instantes de tempo após o início da mídia  $a$ . A Tabela 4.3 apresenta as relações síncronas definidas no modelo SIMM, suas respectivas representações gráficas e descrições.

Tabela 4.3: Relações Temporais do Modelo Hiperídia Proposto

Tipo da Relação	Representação Gráfica	Descrição
<i>Starts</i>		Mídias escravas iniciam quando a mídia mestre começa
<i>Starts_Delay</i>		Mídias escravas iniciam com atraso quando a mídia mestre começa
<i>Finishes</i>		Mídias escravas terminam quando a mídia mestre finaliza
<i>Finishes_Delay</i>		Mídias escravas terminam com atraso quando a mídia mestre finaliza
<i>Meet</i>		Mídias escravas iniciam quando a mídia mestre termina
<i>Meets_Delay</i>		Mídias escravas iniciam com atraso quando a mídia mestre termina
<i>Met_By</i>		Mídias escravas terminam quando a mídia mestre inicia
<i>Met_By_Delay</i>		Mídias escravas terminam com atraso quando a mídia mestre inicia
<i>Before</i>		Apresenta as mídias, sequencialmente, com atraso entre elas

## 4.4 Interatividade

As relações assíncronas no modelo são representadas pela classe *Asynchronous*, que é uma especialização da classe *TemporalRelation*, como mostrado no diagrama de classes do modelo de sincronização temporal na Figura 4.3. Elas correspondem somente aos eventos de interação do usuário implementados pela classe *Interactivity* que, por sua vez, é uma especialização da classe *Asynchronous*. Neste caso, a condição do elo é representada pela seleção da mídia mestre da relação. E as ações podem corresponder ao término dos escravos do elo, finalização da própria mídia interativa ou ainda iniciar novas cadeias temporais.

Como exemplo, considere a Figura 4.4 que mostra um cenário de interatividade através da visão temporal. Nela a mídia mestre da relação é a mídia botão (mídia interativa). Quando esta mídia é selecionada, as demais mídias, destacadas em cinza e que representam os escravos da relação, são finalizadas. Além disso, uma nova cadeia temporal é

iniciada representando o começo da mídia destacada em azul. Note que o modelo não suporta outros eventos assíncronos como testes de variáveis ou adaptação de conteúdo e de apresentação.



Figura 4.4: Cenário de interatividade representada na visão temporal

Estas relações podem possuir, além da mídia mestre e das escravas, uma lista de cadeias temporais, implementada pelo atributo *temporalChainList*, a serem iniciadas após o evento de interatividade ocorrer. Neste caso, o elo temporal causal pode possuir uma ação composta pelos papéis *start* e *stop*. O primeiro papel é associado somente à lista das cadeias temporais. Ou seja, uma nova cadeia temporal é gerada para apresentar as novas mídias que irão ser iniciadas no documento hipermídia representando, assim, a nova sequência de eventos síncronos derivada do evento de interatividade conforme a definição em [34].

No caso do elo de interatividade, a lista de mídias escravas representa os itens de mídia da cadeia temporal à qual a mídia selecionada pertence que terão sua apresentação terminada após a ocorrência do evento de interação do espectador. Em relação à mídia mestre, ela representa a mídia interativa, ou seja, a mídia que o espectador deverá selecionar ou interagir através de um tecla de um dispositivo de entrada (ex.: controle remoto, teclado, mouse etc) para disparar as ações associadas à relação de interatividade.

Além disso, a classe *Interactivity* possui o atributo *interactivityKey* referente à tecla que deve ser pressionada para selecionar a mídia interativa durante sua apresentação. Ademais, a classe *Asynchronous* possui os atributos *startDelay* e *stopDelay*. O primeiro representa o atraso após o evento de interatividade ocorrer para disparar a ação de início das cadeias temporais listadas em *temporalChainList*, ou seja, iniciar a mídia mestre dessas cadeias temporais. Já o segundo atributo indica o atraso para disparar a ação de término sobre o evento de apresentação das mídias escravas da relação de interatividade contidas na lista de mídias escravas (*slaveMediaList*).

Quando uma mídia mestre é associada a um elo de interatividade, a relação somente está disponível enquanto este item de mídia estiver sendo apresentado. Assim, caso o

espectador pressione a tecla especificada pelo elo para interagir com a mídia mestre em um momento em que ela não está sendo apresentada, as ações da relação não serão disparadas. Em outras palavras, a duração da mídia interativa define a janela de oportunidade de interação.

O modelo proposto oferece uma forma simplificada de criar relações temporais causais suportando o evento de interatividade quando comparado a modelos mais complexos e expressivos baseados em eventos. Estas relações definidas facilitam a especificação da sincronização temporal entre os itens de mídia por autores que não possuem conhecimento de programação quando o modelo proposto é utilizado na implementação de um editor gráfico para autoria de documentos hipermídia, como será visto no Capítulo 5.

## 4.5 Características de Apresentação

As características de apresentação de um nó em um modelo hipermídia especificam a aparência que o item de mídia possui ao ser exibido durante a execução do documento multimídia. Elas são representadas, no modelo hipermídia proposto, pela classe *PresentationProperty* conforme a Figura 4.5. Esta classe possui relacionamentos de agregação com as seguintes entidades: *PositionProperty*, *SizeProperty*, *CropProperty*, *TextStyleProperty*, *StyleProperty* e *LevelProperty*. Desta maneira, ela possui atributos referenciando cada uma dessas entidades responsáveis por armazenar as informações relativas às propriedades de apresentação como tamanho, estilo e posição.

Para o atributo do tipo *PositionProperty*, o modelo especifica as propriedades *left*, *top*, *right* e *bottom* que devem ser definidas em pixels ou em porcentagem relativa à largura e altura da tela onde o documento hipermídia é apresentado. Além desses atributos, ainda é especificado para a entidade de propriedades de posição o atributo *orderZ* definido em número inteiro especificando o plano no eixo Z em relação à tela onde o nó será exibido, permitindo definir quais mídias serão apresentadas sobre outras no caso de regiões sobrepostas. Camadas com um valor para *orderZ* maior são exibidas acima de camadas com valor menor.

A classe *SizeProperty* contém os atributos *width* e *height* que devem ser especificados em pixels ou em porcentagem relativa à largura e altura da tela. A Figura 4.6 ilustra os atributos de posicionamento e tamanho discutidos. Eles são aplicados a mídias visuais (imagem, vídeo e texto).

Esta classe também oferece o atributo *aspectRatio* para definir a proporção de tela.

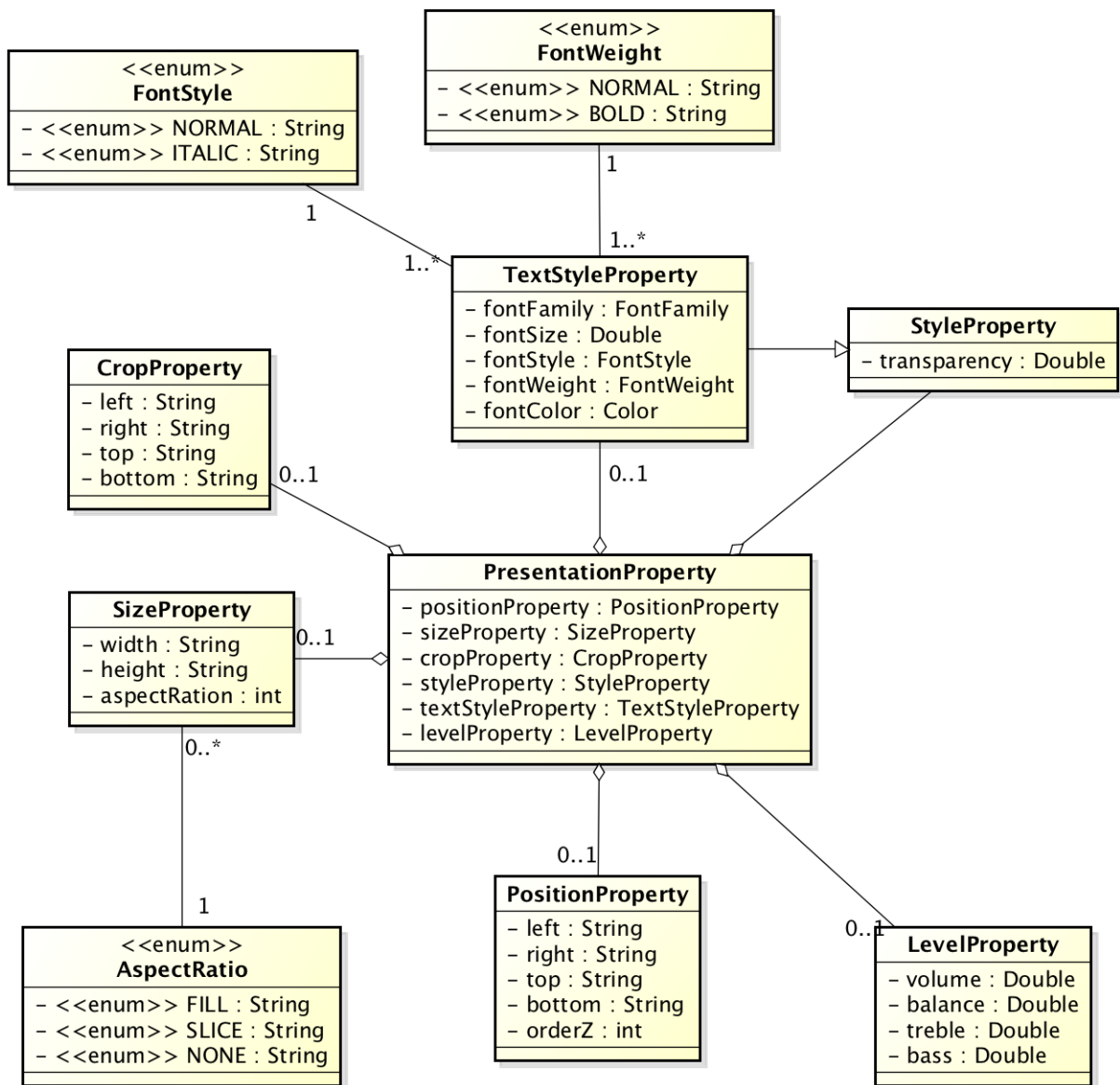
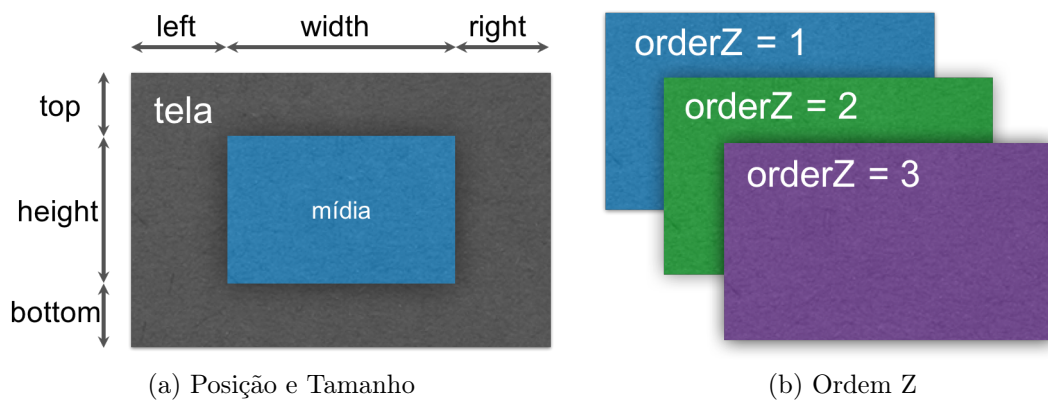


Figura 4.5: Modelo das propriedades de apresentação

Figura 4.6: Propriedades das classes *PositionProperty* e *SizeProperty*



Esta propriedade pode assumir os valores *Fill* e *Slice*. O primeiro redimensiona a mídia para a mesma preencher toda a região da tela, distorcendo-a caso necessário. Esta propriedade está ilustrada na Figura 4.7b. *Slice* define o tamanho da mídia para ela preencher toda tela, porém mantendo sua proporção e omitindo parte da imagem, se necessário. A Figura 4.7c apresenta como uma mídia se comporta quando se aplica o valor *Slice* para a proporção de tela.



Figura 4.7: Proporções de tela do modelo, retirada de [10]

Outra classe que representa característica de apresentação da mídia no modelo proposto é a *CropProperty*. Ela define um recorte da dimensão original da mídia através da especificação dos seguintes atributos: *left*, *right*, *top* e *bottom*. Todos podem ser especificados em pixels em relação às bordas do conteúdo da mídia. A Figura 4.8 ilustra como esses atributos realizam o recorte de uma imagem.

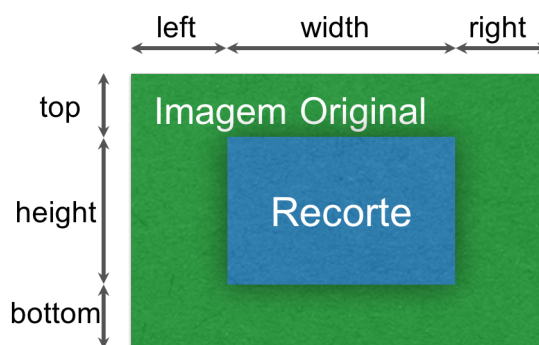


Figura 4.8: Propriedade de recorte

A propriedade de transparência das mídias também é definida no modelo proposto e ela é representada pelo atributo *transparency* da classe *StyleProperty*. Para as mídias do tipo texto, ainda existem as propriedades referentes ao tipo de fonte, tamanho da fonte, estilo da fonte (normal ou itálico), espessura da fonte (normal ou negrito) e cor da fonte. Elas são representadas pela classe *TextStyleProperty*, que é uma especialização da classe

*StyleProperty*, sendo assim, o atributo de transparência também é aplicável às mídias textuais.

Para as mídias do tipo áudio e vídeo, o modelo define os atributos *volume*, *balance*, *treble* e *bass* na classe *LevelProperty* relativos ao volume, equilíbrio do som, nível de agudo e grave respectivamente. Desta forma, o modelo hipermídia proposto permite definir as principais propriedades de apresentação de um item de mídia como enunciado no Capítulo 2 de [38].

## 4.6 Sincronização Espacial

A sincronização espacial também é oferecida pelo modelo, permitindo que itens de mídia de um documento hipermídia sejam posicionados relativos a outros. Para isso, as seguintes entidades são definidas pelo modelo SIMM conforme a Figura 4.9. Os tipos de relações descritos nesta seção foram baseados nas restrições espaciais discutidas em [43].

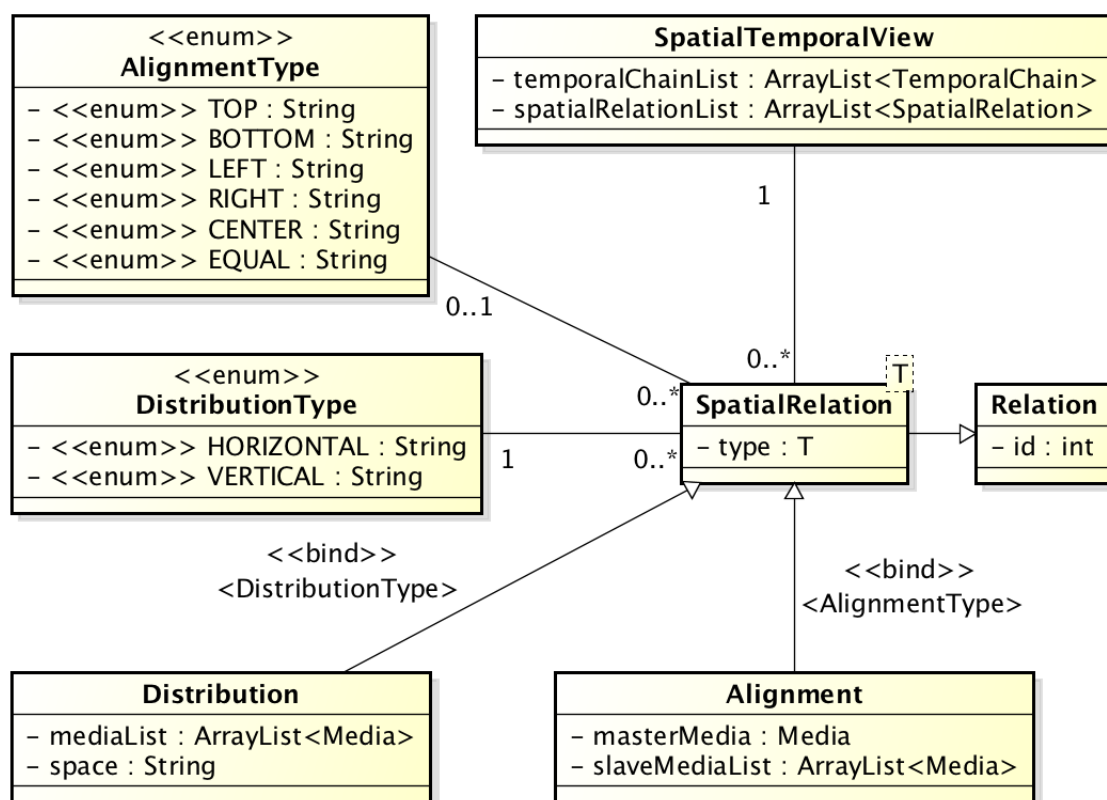


Figura 4.9: Modelo da sincronização espacial

A classe *SpatialRelation* representa as relações espaciais no modelo proposto. Esta classe possui o atributo *type*, o qual é parametrizado a fim de assumir diferentes valores de acordo com o tipo da relação, conforme representado na Figura 4.9 pelo estereótipo *bind*.

O elo espacial pode ser classificado em *Alinhamento* ou *Distribuição* representados pelas classes *Alignment* e *Distribution* respectivamente. Ambas as classes são especializações da classe *SpatialRelation*.

A primeira classificação, Alinhamento, possui uma mídia mestre simbolizada pelo atributo *masterMedia*. O conceito de mestre nas relações espaciais é similar ao especificado na Seção 4.3. Isto é, a mídia mestre define o comportamento, neste caso, espacial das mídias escravas associadas à relação.

**Definição 3.** *Uma mídia mestre ou dominante, em uma relação espacial, é um nó cuja posição espacial na tela de exibição da aplicação multimídia é utilizada, relativamente, para posicionar outros itens de mídia na região da tela.*

Além da mídia mestre, este tipo de relação espacial possui uma lista de mídias denominadas escravas representadas pelo atributo *slaveMediaList*. Estas mídias possuem seus respectivos posicionamentos espaciais definidos relativamente ao mestre da relação espacial, da qual elas participam, de acordo com o tipo de alinhamento estabelecido para o elo.

**Definição 4.** *Uma mídia escrava, em uma relação espacial, é um nó cuja posição espacial é definida relativamente à posição do seu mestre.*

No caso de um alinhamento, o atributo *type*, parametrizado da classe *SpatialRelation*, pode assumir os seguintes valores, conforme a classe *AlignmentType* enumera: *TOP*, *BOTTOM*, *LEFT*, *RIGHT*, *CENTER* e *EQUAL*. A Tabela 4.4 apresenta a descrição e a representação gráfica para cada tipo de alinhamento.

O outro tipo de relação espacial estabelecido no SIMM é referente à classe *Distribution*. Neste caso, os conceitos de mestre e escravo, os quais são utilizados nas relações de alinhamento, não são aplicáveis para as distribuições. A classe *Distribution* apresenta dois atributos: *mediaList* e *space*. O primeiro corresponde à lista de mídias que participam da relação. O segundo representa o espaço que deve ser utilizado entre as mídias integrantes da distribuição. Para este tipo de relação espacial, o atributo *type* de *SpatialRelation* assume os valores *HORIZONTAL* e *VERTICAL* de acordo com a classe *DistributionType*. A Tabela 4.5 apresenta a descrição e a representação gráfica para cada tipo de distribuição.

Tabela 4.4: Relações Espaciais de Alinhamento do Modelo Hipermídia Proposto

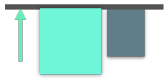
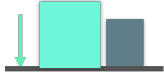





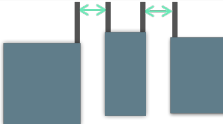
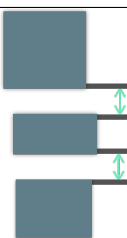
Tipo do Alinhamento	Representação Gráfica	Descrição
<i>TOP</i>		As mídias escravas são alinhadas pelo topo da mídia mestre
<i>BOTTOM</i>		As mídias escravas são alinhadas pela base da mídia mestre
<i>LEFT</i>		As mídias escravas são alinhadas pela borda esquerda da mídia mestre
<i>RIGHT</i>		As mídias escravas são alinhadas pela borda direita da mídia mestre
<i>CENTER</i>		As mídias escravas são alinhadas, horizontalmente, pelo centro da mídia mestre
<i>MIDDLE</i>		As mídias escravas são alinhadas, verticalmente, pelo centro da mídia mestre
<i>EQUAL</i>		As mídias escravas passam a ocupar a mesma área da mídia mestre

Tabela 4.5: Relações Espaciais de Distribuição do Modelo Hipermídia Proposto

Tipo da Distribuição	Representação Gráfica	Descrição
<i>HORIZONTAL</i>		As mídias são distribuídas igualmente na horizontal
<i>VERTICAL</i>		As mídias são distribuídas igualmente na vertical

## 4.7 Limitações do Modelo SIMM

O modelo SIMM possui limitações quando comparado ao NCM, um modelo mais complexo e expressivo. No modelo proposto, não são oferecidos nós de composição e *switches* que permitem, respectivamente, especificar uma estrutura lógica de um documento hipermídia e definir um conjunto de nós alternativos que são selecionados em tempo de execução de

acordo com regras definidas durante a fase de autoria, o que oferece suporte à adaptação de conteúdo. Outro ponto que o SIMM não abrange é a adaptação de apresentação, onde as características de apresentação de itens de mídia são definidas durante a execução do documento.

Âncoras de propriedades também não são contempladas pelo modelo proposto, assim como as de conteúdo. A classe *TimeSegment* apresentada na Seção 4.2 não é considerada como âncora de conteúdo, pois ela não permite definir trechos de mídias discretas como imagens e textos.

O SIMM não oferece suporte a testes de variáveis, os quais permitem definir relações temporais causais que levam em conta valores de variáveis para a decisão do disparo de suas ações. Consequentemente, o modelo não considera os eventos de atribuição como descritos para o modelo NCM na Seção 2.4.1.

As relações de restrições também não são contempladas no modelo proposto. No que diz respeito à relação causal do SIMM, ela sempre possui uma condição e ação simples estabelecidas pela definição do mestre, escravos e tipo do elo, diferentemente, do modelo NCM onde as relações causais são multipontos. Entretanto, para um mesmo mestre, podem ser definidas outras relações causais fazendo com que uma mesma condição possa disparar mais de um tipo de ação fazendo uso do modelo proposto.

Por exemplo, considere as seguintes especificações de elos utilizando o modelo SIMM. O primeiro é do tipo *Starts*, conforme definido na Tabela 4.3, o qual estabelece que os itens escravos devem iniciar suas respectivas apresentações quando o mestre começar. O segundo elo do tipo *Met By* é definido tendo como mestre o mesmo item utilizado na primeira relação. Desta forma, quando este mestre iniciar sua apresentação, os escravos da primeira relação começarão e os associados ao segundo elo terminarão suas respectivas apresentações, ou seja, um item mestre está associado ao papel de condição *onBegin* e aos papéis de ação *start* e *stop*.

Ademais, o SIMM não permite definir efeitos de animações, os quais alteram valores das características de apresentação do item de mídia ao longo da execução do documento, como também especificar navegação por teclas para alterar o foco para diferentes objetos de mídia. Este último recurso é oferecido pelo modelo NCM através de atributos do elemento descritor.

## 4.8 Conclusão

Este capítulo introduziu o modelo de documentos hipermídia proposto neste trabalho, o SIMM (*Simple Interactive Multimedia Model*). O modelo apresentado neste trabalho define um conjunto de entidades simplificado em relação à hierarquia de classes presente no modelo NCM, conforme é apresentado no Capítulo 2. Desta forma, o SIMM possui um modelo de sincronização temporal com um nível de abstração relativamente mais alto do que o NCM, onde as relações temporais são definidas apenas através da especificação de uma mídia mestre, de uma lista de escravos, do tipo da relação e, opcionalmente, do valor de atraso. Esta simplificação fornece um modelo que serve como base para implementação de um sistema de autoria hipermídia que ofereça relações temporais simples, baseadas nas relações de Allen [27] e razoavelmente expressivas para representar classes essenciais de documentos hipermídia.

Ademais, o capítulo também discutiu a maneira como o modelo SIMM suporta a definição de relações de interatividade e apresentou as classes do modelo SIMM que representam as características de apresentação dos itens de mídia durante a execução do documento hipermídia. Ao final, foram discutidos o modelo de sincronização espacial do SIMM e suas limitações em relação ao modelo NCM.

O próximo capítulo irá apresentar o editor gráfico da visão espaço-temporal para autoria de documentos hipermídia proposto neste trabalho, STEVE (*Spatio-Temporal View Editor*), discutindo suas funcionalidades, arquitetura, interface gráfica e como a consistência temporal é mantida ao longo da edição da sincronização temporal da aplicação hipermídia.

# Capítulo 5

## STEVE

Este capítulo apresenta o editor gráfico da visão espaço-temporal para autoria de documentos hipermídia proposto neste trabalho, o qual é chamado STEVE, *Spatio-Temporal View Editor*. A primeira seção do capítulo discute, de uma maneira geral, o que as ferramentas de criação e edição de aplicações multimídia oferecem aos usuários. Além disso, ela discute sobre o paradigma de autoria utilizado por estas ferramentas, apontando suas vantagens e desvantagens. Também, apresenta os objetivos e características gerais do editor proposto neste trabalho. Na Subseção 5.1.1, a arquitetura da ferramenta é apresentada, mostrando suas entradas e saídas. A interface gráfica e a barra de menu do editor são discutidas na Seção 5.1.2. O repositório de mídias presente na ferramenta é descrito na Seção 5.2.

Na Seção 5.3, a visão temporal oferecida por STEVE é apresentada, discutindo, primeiramente, sua interface gráfica, seu objetivo e algumas de suas funcionalidades. Em seguida, a manipulação das mídias é estudada em detalhes na Seção 5.3.1, apresentando como a adição e remoção de mídias são realizadas pelo usuário, além de descrever a alteração da duração e da disposição dos itens de mídias na cadeia temporal. Na Seção 5.3.2, é apresentado como STEVE implementa a sincronização temporal através do modelo hipermídia, SIMM, proposto no Capítulo 4. Logo após, a Seção 5.3.3 apresenta o suporte à interatividade dado pelo editor proposto. Por fim, na Seção 5.3.4, são apresentados tratamentos dados a cada edição realizada pelo autor que pode comprometer a consistência temporal do documento. Os tipos de edição tratados correspondem à criação de relações temporais síncronas, ao reposicionamento de mídias, à remoção de mídias e cadeias temporais e à alteração da duração dos itens de mídias.

A visão espacial é discutida na Seção 5.4, apresentando sua interface gráfica e como ela é integrada à visão temporal, de forma a permitir a prévia das aplicações hipermídia.

Ademais, esta seção mostra os recursos oferecidos por STEVE para possibilitar que o autor possa editar as características de apresentação dos itens de mídia.

## 5.1 Visão Geral

A ferramenta proposta neste trabalho é chamada de STEVE, Spatio-temporal View Editor. Ela permite que autores sem conhecimento em linguagens ou modelos multimídia sejam capazes de criar apresentações multimídia interativas através da visão espaço-temporal, podendo definir eventos de interatividade e relações temporais causais entre os itens de mídia. Dessa maneira, a ferramenta lida com as limitações do paradigma *timeline* fazendo uso do modelo de sincronização temporal baseado em eventos.

STEVE oferece aos usuários em sua interface gráfica relações temporais condicionais pré-definidas, baseadas nas treze relações temporais entre intervalos de Allen [27], de forma a facilitar a sincronização das mídias do documento por usuários leigos. Esta funcionalidade do editor será apresentada na Seção 5.3. Além disso, o editor proposto permite que o autor especifique características de apresentação dos itens de mídia e verifiquem como as mídias serão apresentadas durante a execução do documento. Para isto, a ferramenta disponibiliza a prévia da aplicação multimídia mostrando como e onde os componentes do documento serão apresentados ao longo do tempo. A visão espacial das aplicações oferecida por STEVE será detalhada na Seção 5.4.

A definição de eventos de interatividade, onde o espectador interage com os itens de mídia do documento, também é suportada pelo editor gráfico proposto. O autor necessita selecionar a mídia a qual deseja tornar interativa e informar a tecla/botão que o espectador deve pressionar para interagir com a mídia, além das ações que serão executadas após a ocorrência do evento interativo. Para que o autor possa visualizar como a aplicação se comportaria caso a interatividade ocorresse, a ferramenta permite que o autor informe um valor de instante de tempo hipotético no qual a interação ocorreria. Assim, o editor exibe uma cadeia temporal resultante da ocorrência da interatividade. O suporte à interatividade será discutido em detalhes na Subseção 5.3.3.

O editor gráfico proposto exporta as aplicações hipermídia criadas para duas linguagens multimídia padrões: NCL e HTML5. Dessa forma, as aplicações criadas no STEVE podem ser executadas em diversas plataformas como TV digital, IPTV, smartphones e etc. O suporte a estas linguagens será estudado no Capítulo 6.

STEVE é implementado através da tecnologia JavaFX [15] para oferecer aos usuários



uma rica interface gráfica que será mostrada na Subseção 5.1.2. Já na Subseção 5.1.1, a arquitetura de fluxo de dados do editor será analisada apresentando as entradas e saídas da ferramenta. Na subseção a seguir, iremos citar os principais requisitos funcionais levantados após o estudo dos trabalhos relacionados.

### 5.1.1 Arquitetura

A arquitetura de fluxo de dados do editor é apresentada na Figura 5.1. O diagrama de fluxo de dados tem três entradas: novos projetos STEVE, arquivos de projeto do STEVE preexistentes e documentos NCL. A entrada de novo projeto corresponde ao início de uma nova aplicação hipermídia no STEVE. No caso de projetos preexistentes, o usuário inicia a ferramenta e carrega um arquivo de projeto STEVE que representa a aplicação multimídia interativa implementada em objetos Java conforme o modelo hipermídia proposto no Capítulo 4.

A partir do modelo SIMM, o sistema pode gerar um documento hipermídia em NCL, transformando as entidades do modelo SIMM em elementos da linguagem NCL. Para escrever o documento NCL, o sistema faz uso da API aNa [60], que oferece um framework para especificar uma aplicação NCL através de objetos Java. O sistema também pode exportar a aplicação para um documento especificado em HTML 5 [14], fazendo uso da ferramenta NCL4Web [61], a qual usa folha de estilo e bibliotecas Javascript para transformar documentos NCL em aplicações HTML 5. Além disso, a aplicação aberta no sistema pode ser salva no formato de arquivo do projeto STEVE para futuras edições. Ambas as ferramentas estão destacadas em lilás na Figura 5.1.

Quando o sistema recebe um documento NCL como entrada, ele o transforma num modelo de grafo hipermídia temporal, *Hypermedia Temporal Graph* - HTG [39]. Em seguida, o HTG do documento de entrada é mapeado para o plano de apresentação, outra estrutura de dados proposta em [39], representando um conjunto de transições de eventos disparados sobre os componentes da aplicação e cada uma destas transições possuindo seus instantes de tempo de execução. Posteriormente, o plano de apresentação permite determinar os instantes de tempo em que os itens de mídia do documento são exibidos. Ou seja, permite construir as cadeias temporais da aplicação fornecendo, para cada item de mídia, o instante de início e término do evento de apresentação. Este processo de transformação do documento NCL nestas três estruturas de dados, destacadas em laranja na Figura 5.1, foi implementado num trabalho próprio anterior [54]. Com a cadeia temporal gerada, ela é mapeada para as entidades do modelo hipermídia do STEVE. Mais

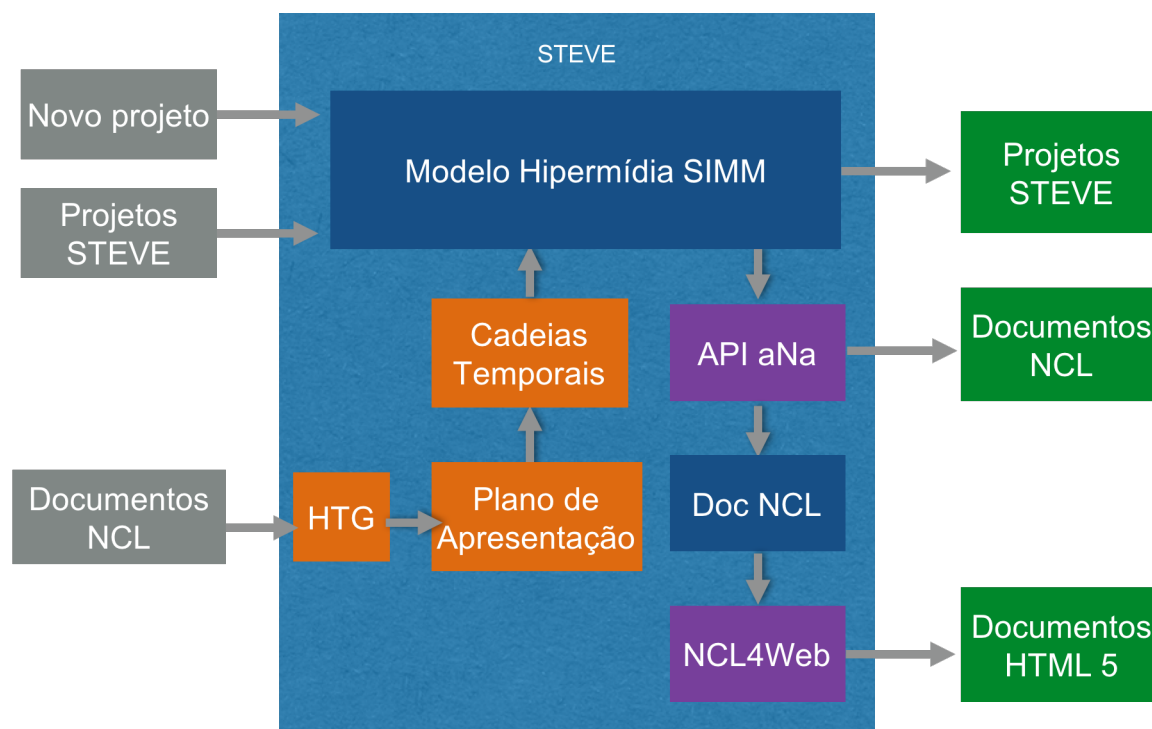


Figura 5.1: Diagrama de Fluxo de Dados do STEVE

informações sobre o suporte à leitura de documentos NCL pelo editor proposto serão dadas no Capítulo 6.

### 5.1.2 Interface Gráfica

A interface gráfica da ferramenta proposta foi projetada com base nas ferramentas comerciais apresentadas no capítulo de trabalhos relacionados. Em geral, elas oferecem, em conjunto, a visão temporal e espacial das apresentações multimídia conforme mostrado na Seção 3.1. Estes editores de âmbito comercial fornecem estas visões integradas através da linha do tempo e da prévia da aplicação, permitindo, ainda, definir propriedades de estilo, posição e tamanho das mídias.

Esse leiaute de interface é bastante adotado no meio comercial, visto que é bem aceito entre os usuário, principalmente, entre os que não possuem conhecimento de modelos e linguagens de autoria multimídia, devido à facilidade que é oferecida na compreensão do comportamento espacial e temporal das aplicações multimídia em construção.

Desta forma, a interface gráfica do STEVE foi desenhada contendo três regiões: o repositório de mídias, a visão temporal, composta por uma barra de botões e a linha do tempo, e a visão espacial possuindo o painel de propriedades de apresentação das mídias e a tela de prévia da aplicação hiperímia. Cada uma delas será apresentada em detalhes

nas Seções 5.2, 5.3 e 5.4 respectivamente, mostrando suas funcionalidades e como o usuário interage com elas. Essas regiões podem ser visualizadas na Figura 5.2.

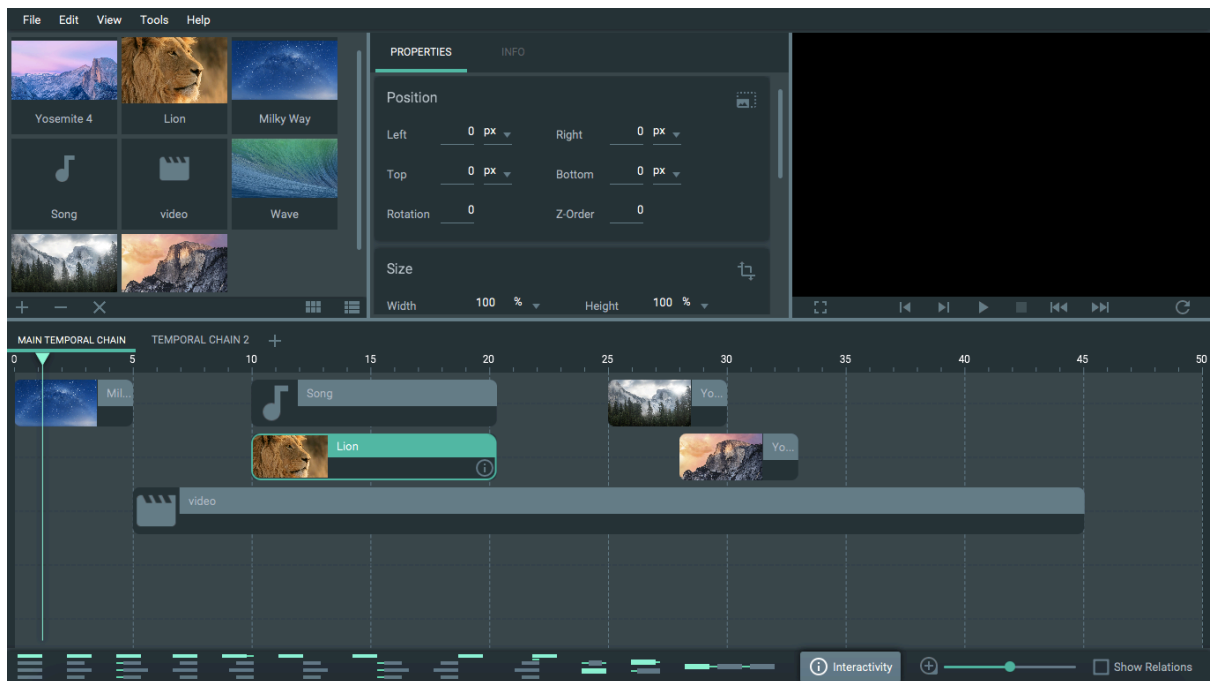


Figura 5.2: Interface Gráfica do STEVE

O estilo da interface gráfica do STEVE foi idealizada utilizando os princípios da linguagem visual *Material Design* da Google [23] [13]. Cada componente gráfico de tela (botões, campos, caixas de diálogo etc) e estilo do editor como ícones [24] e cores foram implementados com base nas recomendações desta linguagem a fim de proporcionar ao usuário uma interface simples e limpa enfatizando as principais funcionalidades da ferramenta. Para facilitar a implementação desses requisitos e a manipulação de mídias e gráficos, a ferramenta proposta utiliza em sua implementação a API JavaFX 8 [15].

Além disso, a construção de uma interface gráfica clara e objetiva visa garantir a qualidade da usabilidade da ferramenta e da experiência do usuário. A usabilidade de um sistema, conforme apresentado em [11], é definida como se segue:

**Definição 5.** *Em que medida um sistema, produto ou serviço pode ser usado por usuários específicos para atingir determinados objetivos de forma efetiva, eficiente e satisfatória em um contexto de uso.*

No caso do termo experiência do usuário, ele possui diversas definições [52] entre especialistas da área de interação humano-computador. Entretanto, a definição, apresentada a seguir, é especificada pela ISO [11] e bem aceita [52] tanto no âmbito industrial quanto acadêmico.

**Definição 6.** *A experiência do usuário é definida como sendo as respostas e percepções do usuário que resultam do uso ou do uso antecipado de um produto, sistema ou serviço.*

As definições em [11] de usabilidade e experiência do usuário podem ser bem próximas ao afirmar que a experiência do usuário inclui todos os comportamentos e percepções do usuário antes, depois e durante o uso [30]. Porém, a experiência do usuário foca na experiência de um indivíduo em relação à sua percepção, preferências, emoções e respostas físicas e psicológicas contrastando com a visão de eficácia, eficiência e satisfação de um grupo de usuários, conforme discutido em [31] onde um dos objetivos é esclarecer que o componente de satisfação da usabilidade inclui aspectos da experiência do usuário.

Os critérios e métodos utilizados para realizar os testes de usabilidade e da experiência do usuário do STEVE como também seus resultados serão apresentados no Capítulo 7.

Na Figura 5.2, também pode ser observada a barra de menu do STEVE. Em *File*, existem as opções: abrir projeto, salvar projeto, novo projeto, fechar projeto e sair da ferramenta, além das opções de importar um documento NCL e exportar projetos do STEVE para documentos NCL ou HTML5 como será visto no Capítulo 6.

No menu *Edit*, existem as seguintes opções ainda indisponíveis na versão 1.0-0 para refazer e desfazer a última ação, copiar, colar e recortar um item de mídia. Ainda nesse menu, a ferramenta oferece opções para selecionar todos os itens de mídia de uma cadeia temporal, deletar um item de mídia selecionado e limpar uma cadeia temporal. Esta última funcionalidade permite que o usuário remova todos os itens de mídia e relações presentes na cadeia.

Em relação ao menu *View*, existem opções para ocultar cada uma das regiões da interface gráfica: repositório de mídias, visão temporal, o painel de propriedades das mídias e a tela de prévia da aplicação hipermídia. Em *Tools*, existem as opções para realizar a simulação da interatividade e selecionar o idioma desejado (disponível em inglês e português) do STEVE. E na última opção da barra de menu, *Help*, existem duas opções: uma para exibir um manual do usuário da ferramenta (ainda não disponível na versão 1.0-0) e outra para mostrar a janela com informações do editor como versão e autoria.

## 5.2 Repositório de Mídias

STEVE possui um repositório de mídias onde são armazenados todos os arquivos de mídia que o usuário importou para utilizá-los na criação da sua aplicação multimídia interativa.

Isto possibilita que o autor organize suas mídias em um único lugar facilitando e agilizando na seleção dos itens de mídia para o documento hipermídia. Será estudado, mais à frente, como o repositório funciona, de forma integrada, com as visões temporal e espacial. A interface do repositório de mídias pode ser visualizada na Figura 5.3.

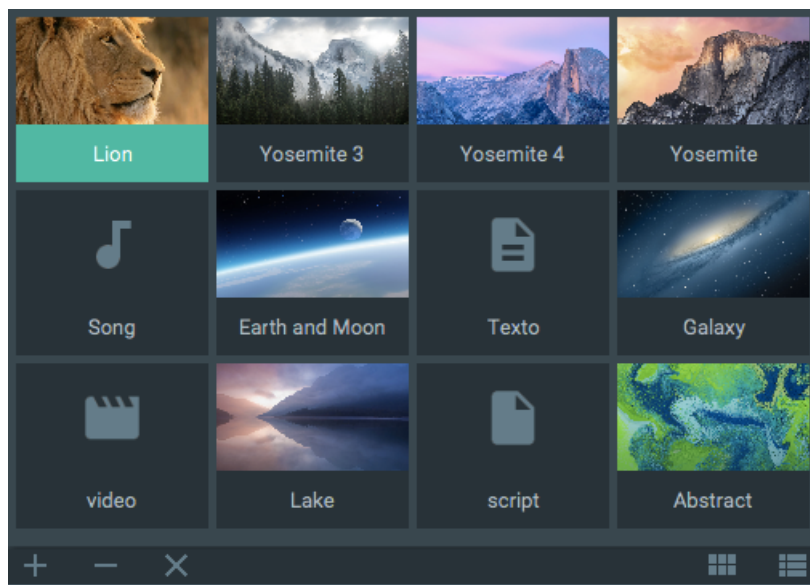


Figura 5.3: Repositório de Mídias do STEVE. Formato em Ícones.

Nesta figura, pode-se notar que existe uma barra de botões na parte inferior do repositório. Ela contém cinco botões no total. Três deles estão alinhados à esquerda: o botão de adicionar/importar arquivos de mídia para o ambiente do editor, de remover uma mídia selecionada no repositório e outro botão para limpar o repositório apagando todas as mídias já importadas pelo usuário.

Os outros dois botões estão alinhados à direita e correspondem à funcionalidade de alternar o leiaute de exibição das mídias. O usuário pode optar por exibir as mídias do repositório de duas formas: em ícones e em lista. Na primeira opção, as mídias são representadas graficamente no repositório como ícones. Para a mídia do tipo imagem, o ícone representado pelo *thumbnail* (versão reduzida da imagem) da mídia; para mídia vídeo, áudio, texto e aplicação, são usados símbolos para representá-los. A Figura 5.3 ilustra esta representação. A forma em lista agrupa os arquivos de mídia por tipo em uma estrutura de árvore. Ela possui cinco nós onde cada um representa um tipo de mídia, o qual pode ser imagem, vídeo, áudio, texto e aplicação. Esta forma de exibição é mostrada na Figura 5.4.

Os arquivos de mídia do tipo aplicação são adequados na representação de códigos que podem ser inseridos em aplicações NCL. Como exemplos, existem os objetos de mídia

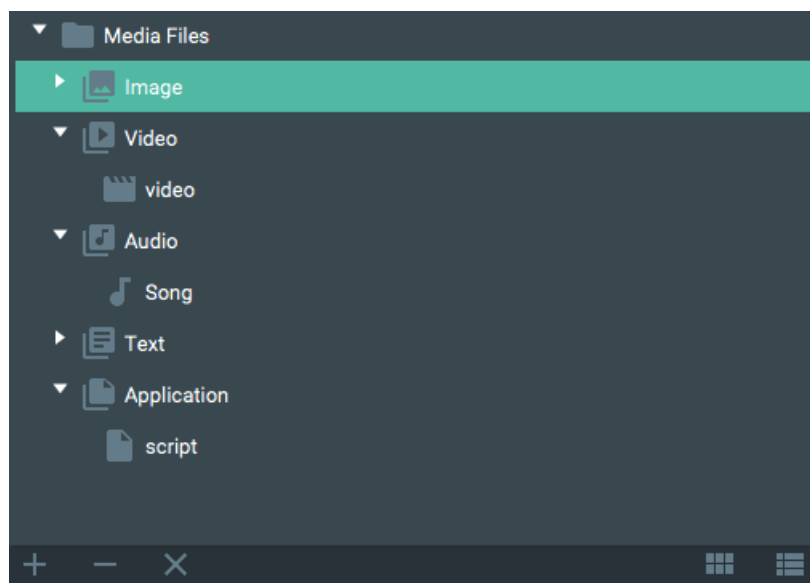
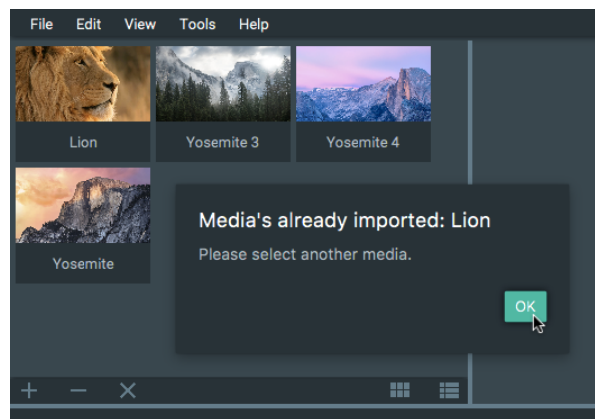


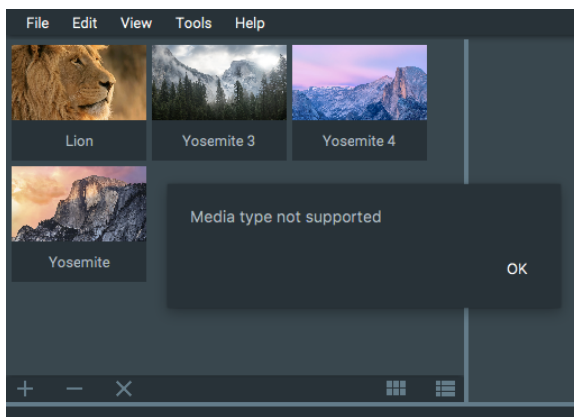
Figura 5.4: Repositório de Mídias do STEVE. Formato em Lista.

do tipo *application/x-ginga-NCL*, que representam um documento especificado em NCL; *application/x-ginga-NCLua*, ou *application/x-ncl-NCLua*, que indicam código Lua [50], linguagem de script para documentos NCL; *application/x-ginga-NCLet*, ou *application/x-ncl-NCLet*, que possuem como conteúdo código imperativo Java, dando mais expressividade aos documentos NCL. Todos esses arquivos de aplicação assumem o mesmo comportamento da máquina de estado associada a um objeto de mídia não-imperativo, conforme apresentado no Capítulo 1. Mais informações sobre o uso desses objetos de mídia de aplicação podem ser encontradas em [62].

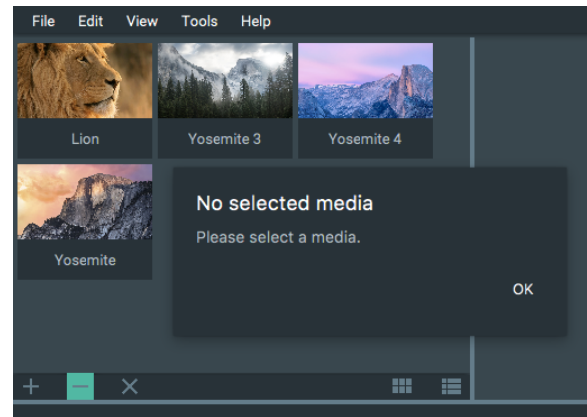
Com o objetivo de orientar os usuários na forma adequada de utilização das funcionalidades do repositório de mídias, o sistema pode exibir janelas com mensagens de alertas. A Figura 5.5a ilustra o alerta que é exibido pelo editor quando o usuário tenta importar um arquivo de mídia já presente no repositório, informando a mídia duplicada e orientando a tentar importar outro arquivo. No caso em que o usuário seleciona uma mídia cujo tipo não é suportado pela ferramenta, é mostrado um alerta informando que o sistema não permite a utilização deste tipo de mídia conforme a Figura 5.5b. Já a Figura 5.5c informa ao usuário que ele deve selecionar uma mídia no repositório para removê-la recomendando que ele escolha uma mídia quando nenhuma é selecionada.



(a) Mídia Já Importada



(b) Mídias Não Suportadas



(c) Nenhuma Mídia Seleccionada

Figura 5.5: Alertas no Uso do Repositório

## 5.3 Visão Temporal

A ferramenta proposta neste trabalho permite que usuários criem suas aplicações hiper-mídia através da manipulação dos itens de mídia ao longo de um eixo do tempo. Esta edição da visão temporal dos documentos oferecida por STEVE faz uso das relações temporais causais definidas no modelo proposto conforme apresentado no Capítulo 4. Desta maneira, esta visão permite que o autor defina relações temporais entre os eventos (de apresentação e seleção) que ocorrem com os itens de mídia do documento a fim de realizar a sincronização temporal da aplicação.

A Figura 5.6 mostra a visão temporal do STEVE, onde neste exemplo existem duas cadeias temporais: *Main Temporal Chain* e *Temporal Chain 2*. A primeira delas é denominada de cadeia temporal principal cujo item de mídia mestre inicia a aplicação hipermídia e as demais são classificadas de cadeias secundárias.

Conforme foi mostrado na Seção 4.3, uma aplicação pode ser composta por diversas

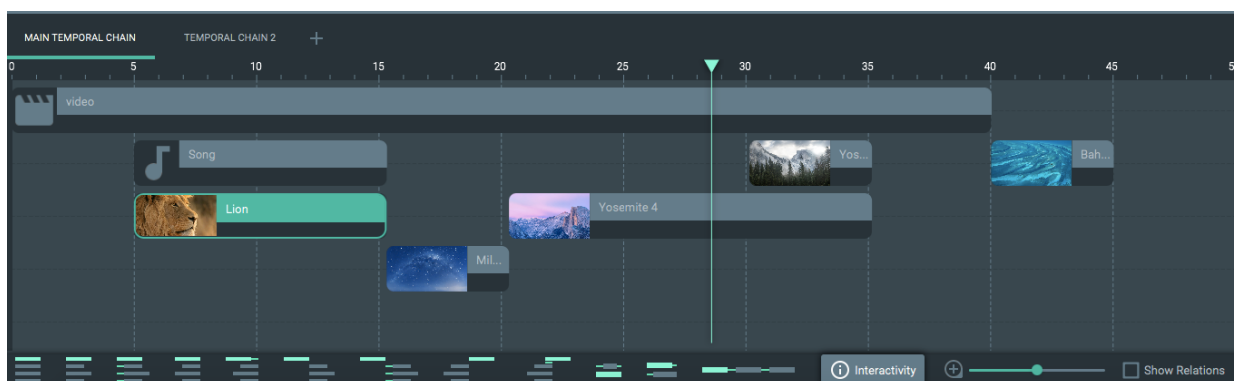


Figura 5.6: Interface Gráfica da Visão Temporal do STEVE

cadeias temporais. Então, o usuário pode criar inúmeras cadeias através da seleção do botão, indicado com um sinal de soma (+) na Figura 5.7, para adicionar novas abas, cada uma representando uma nova cadeia. Para deletar uma cadeia, basta clicar no botão 'x' ao lado do nome da cadeia mostrado também na Figura 5.7. Ainda é possível renomear as cadeias clicando na aba que a representa com o botão direito do mouse e selecionar a opção *Rename*.

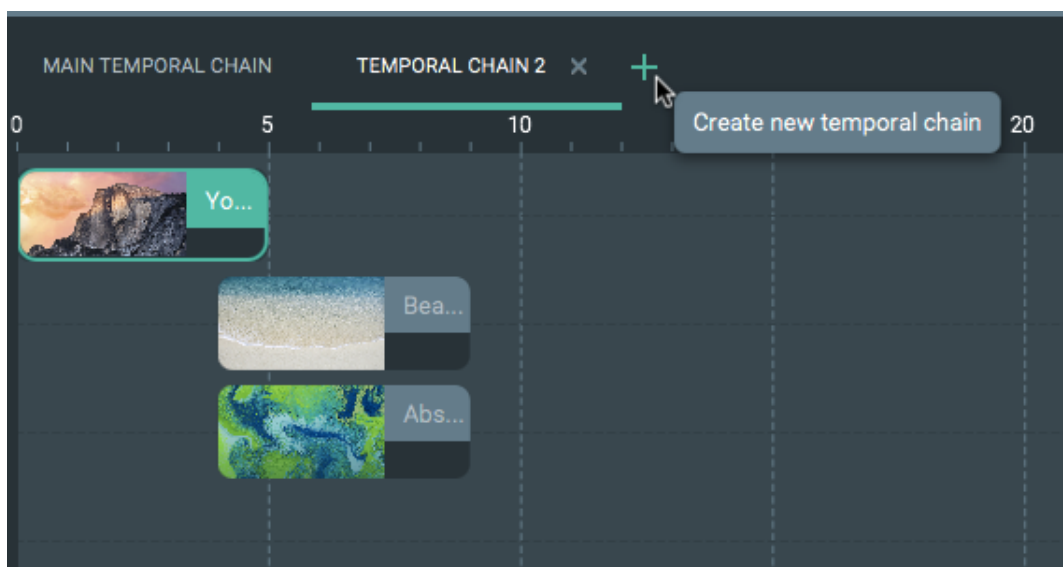


Figura 5.7: Abas para Suportar Inúmeras Cadeias Temporais

Estas cadeias temporais serão associadas umas às outras quando o autor definir um evento de interatividade que relacione uma mídia em uma cadeia com o início de alguma dessas cadeias secundárias. Mais detalhes da funcionalidade de definição da interatividade serão abordados na Subseção 5.3.3.

Por padrão, a unidade de tempo utilizada no eixo da visão temporal é o segundo e o valor do tempo é indicado, explicitamente, a cada 5 segundos conforme a Figura 5.7. O nível de zoom em que são exibidas as mídias na cadeia temporal pode ser ajustado de



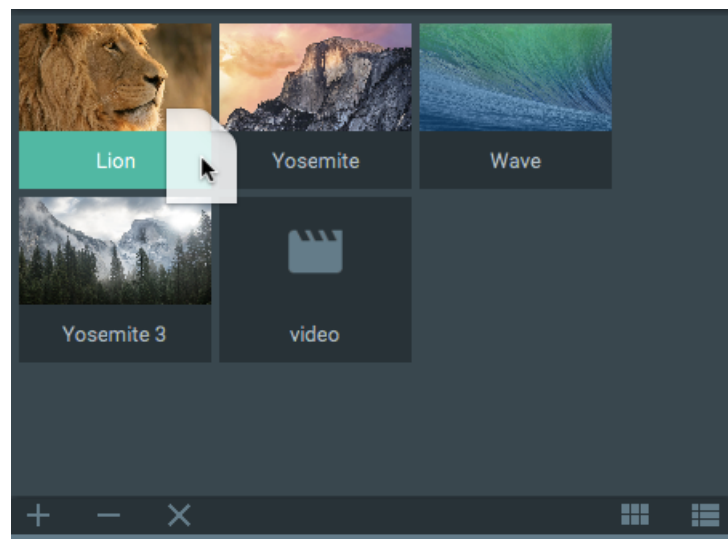
acordo com o usuário através do botão de nível de zoom presente na barra de botões da visão temporal mostrada na Figura 5.6. As outras funcionalidades da barra de botões serão apresentadas nas subseções seguintes.

### 5.3.1 Manipulação das Mídias na Visão Temporal

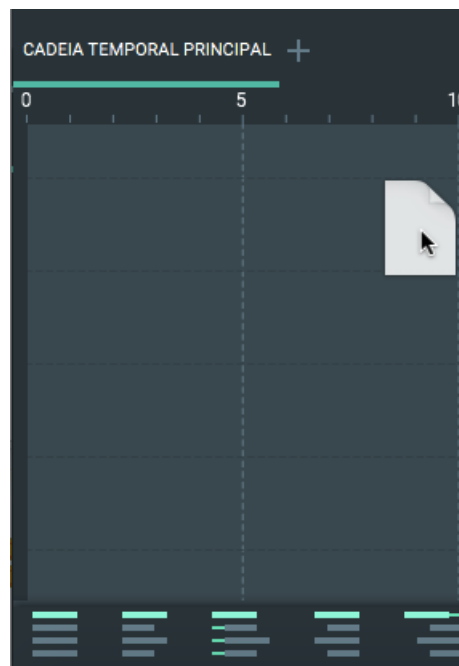
Quando o usuário deseja adicionar itens de mídia à sua aplicação hipermídia, ele deve selecionar uma das mídias armazenadas no repositório de mídia e arrastá-la para a região da cadeia temporal que ele deseja inserir a mídia conforme a Figura 5.8. Nela, é inserida a mídia do tipo imagem *Lion* na cadeia principal. Como esta mídia é a primeira a ser adicionada a esta cadeia, a imagem *Lion* é definida pelo sistema como sendo a mídia mestre da cadeia. Esta mídia é posicionada no instante zero da cadeia temporal independentemente do instante escolhido pelo usuário ao soltar a mídia na visão temporal. Visto que esta cadeia é a principal, essa mídia imagem inicia a aplicação multimídia.

Para as subseqüentes inserções de mídia, o início de apresentação do item de mídia se torna, exatamente, o instante em que a mídia é solta na cadeia temporal. Para facilitar a seleção do instante de tempo no momento em que a mídia é solta na cadeia, o editor exibe uma linha vertical sobre o eixo do tempo indicando o instante de início que o item de mídia irá assumir após o usuário soltá-la na visão temporal. Esta linha indicativa pode ser visualizada na Figura 5.8. Já o instante de término da mídia é definido de acordo com sua duração acrescida do instante de seu início. Para as mídias contínuas (áudio e vídeo), o sistema obtém sua duração natural. Porém, no caso das mídias discretas, o editor define uma duração explícita de 5s.

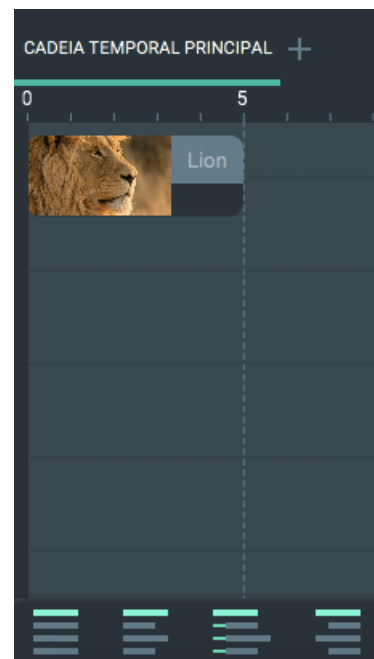
Visando garantir a organização e qualidade na edição do documento, o sistema distribui, automaticamente, as mídias inseridas nas cadeias otimizando o espaço ocupado por elas e facilitando, assim, a edição e visualização da ordem de apresentação do documento pelos autores. A fim de realizar essa otimização, para cada mídia a ser inserida na cadeia temporal, o sistema verifica, sempre, se é possível adicioná-la na primeira linha (de cima para baixo na cadeia temporal) disponível para aquele instante de início e término que o item de mídia irá assumir na cadeia. Para verificar essa disponibilidade, o sistema faz uso das relações de Allen [27] identificando qual a relação temporal existente entre o intervalo de duração da mídia a ser inserida e de cada mídia das linhas da cadeia temporal. Ao longo dessa verificação, o sistema analisa se a relação encontrada é *Before* ou *After*, o que significa que é possível inserir a mídia na linha onde uma dessas relações foi encontrada. Na Figura 5.9a, usuário está inserindo uma mídia no instante 4s, logo após o item de mí-



(a) Mídia selecionada no repositório



(b) Mídia sendo arrastada para a visão temporal



(c) Mídia inserida na visão temporal

Figura 5.8: Adicionando Itens de Mídia à Aplicação Hipermissão

dia *Lion*. Quando o usuário a solta na cadeia temporal, o sistema a adiciona na primeira linha da cadeia. Ela é destacada em verde, conforme a ilustração da Figura 5.9b, onde seu intervalo de duração não sobrepõe qualquer mídia já presente nesta mesma linha.

Para deletar uma mídia, basta o usuário selecionar alguma mídia na cadeia temporal e pressionar a tecla *Delete* ou clicar com o botão direito sobre a mídia desejada e selecionar a opção *Delete Media*. Ademais, o usuário pode remover diversas mídias de uma mesma cadeia simultaneamente, segurando a tecla *Shift* ou ainda *Command* para teclados Apple

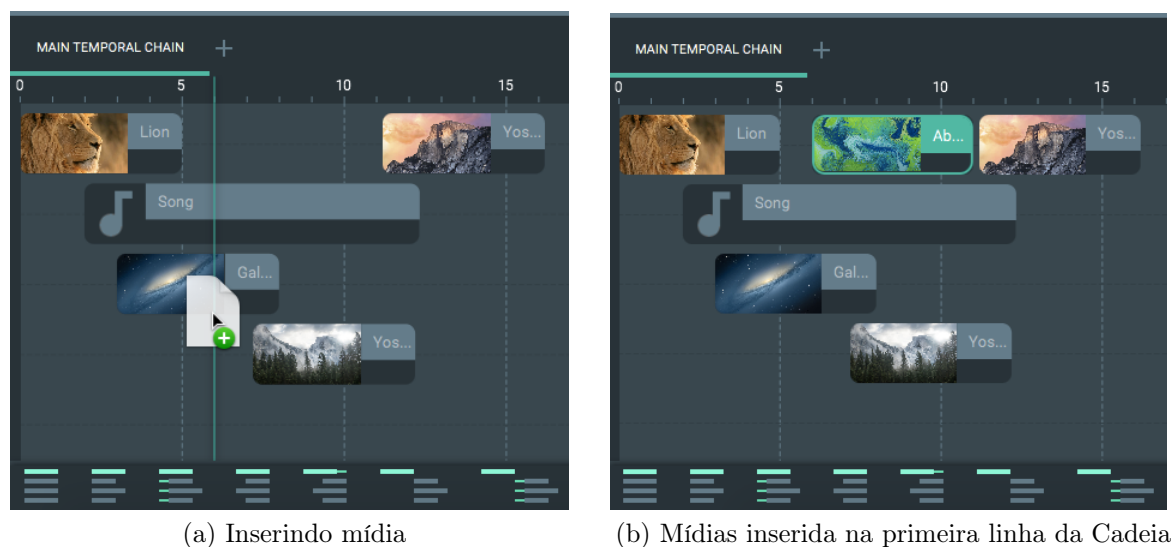


Figura 5.9: Otimização da disposição das mídias na cadeia temporal

enquanto seleciona as mídias e, em seguida, pressionando a tecla *Delete*.

Ao longo da edição do documento, a mídia mestre de uma cadeia pode ser deletada e, caso exista outra mídia na cadeia, o sistema, automaticamente, seleciona como a nova mídia mestre da cadeia, a mídia com menor instante de início presente na cadeia.

Após uma mídia ser inserida na cadeia temporal, ela pode ser arrastada para outro instante de tempo ao longo do eixo e também ter sua duração alterada pelo usuário. Na primeira funcionalidade, basta o autor selecionar uma mídia na cadeia temporal, pressionando e segurando o botão esquerdo do mouse, e arrastá-la para o instante de tempo desejado. O item de mídia é arrastado, somente, ao longo da linha da cadeia à qual ele pertence. Quando o usuário solta o item de mídia, o sistema, automaticamente, o insere na primeira linha da cadeia disponível. Essa otimização da disposição das mídias na visão temporal já foi descrita anteriormente e ilustrada na Figura 5.9.

Para alterar a duração de um item de mídia, o usuário deve posicionar o mouse sobre uma das bordas laterais do retângulo que representa a mídia graficamente na cadeia temporal, pressionar e segurar o botão esquerdo do mouse e arrastá-la aumentando ou diminuindo seu tamanho conforme a duração desejada.

Quando um item de mídia áudio possui uma duração implícita menor que a duração editada pelo usuário, a ferramenta define que o conteúdo da mídia seja repetido após seu término natural até que a duração estabelecida pelo usuário seja alcançada. Para as mídias do tipo vídeo, a ferramenta exibe uma janela, solicitando que o usuário escolha umas das seguintes opções: repetir o conteúdo do vídeo como realizado para as mídias do

tipo áudio ou fixar o último quadro do vídeo na apresentação do documento hipermídia até atingir a duração definida pelo usuário.

Já para os casos onde a duração definida pelo usuário é menor que a implícita da mídia contínua (vídeo ou áudio), STEVE corta o conteúdo das mídias de forma a apresentá-las somente o suficiente para atingir a duração determinada pelo o autor.

Na Subseção 5.3.4, será discutido, em detalhes, o comportamento da ferramenta para manter a consistência das relações temporais causais quando a duração das mídias é alterada existindo dependências entre elas.

### 5.3.2 Sincronização Temporal dos Itens de Mídia

Baseado nas relações temporais condicionais pré-definidas conforme apresentado na Tabela 4.3, STEVE permite que o usuário relacione, temporalmente, os itens de mídia de seu documento hipermídia utilizando elos que associam ocorrências de eventos e ações que alteram o estado de apresentação das mídias. Para isso, a ferramenta apresenta, na barra de botões da visão temporal conforme a Figura 5.6, as relações temporais que o usuário pode utilizar para sincronizar sua aplicação hipermídia temporalmente. A representação gráfica dos ícones dos botões que permitem criar tais relações visam facilitar o usuário a identificar o tipo de elo que deseja definir. A Tabela 5.1 enumera as relações temporais oferecidas pelo STEVE mostrando o símbolo utilizado por cada uma na barra de botões da visão temporal.

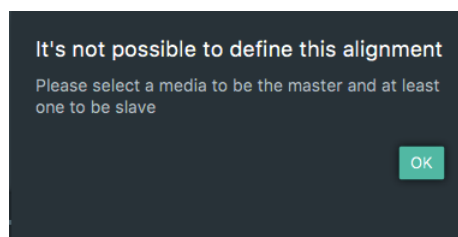
Os ícones dos botões que permitem definir relações no STEVE possuem um único retângulo verde indicando o mestre da relação e outros retângulos cinzas representando os escravos do elo temporal. Isto indica que o usuário pode selecionar vários itens de mídia em uma cadeia temporal para serem escravos de uma relação, exceto para as relações *During* e *Overlaps* como será discutido mais à frente.

Para definir uma relação temporal no STEVE, o usuário deve selecionar uma mídia como mestre, destacada com borda e fundo verde, e pelo menos uma como escrava, destacada somente com a borda verde. A primeira mídia selecionada pelo usuário é definida como a mestre da relação e as demais selecionadas são definidas como escravas. Caso não seja selecionado o número necessário de mídias para definir uma relação, o sistema mostrará alertas conforme a Figura 5.10. Em seguida, o usuário deve pressionar em um dos botões para definir a relação temporal desejada.

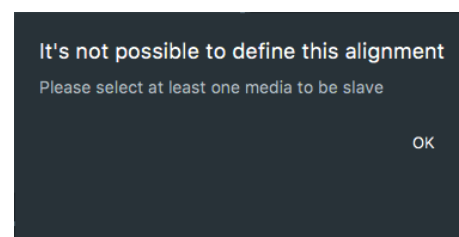
Na Figura 5.11, a relação *Equals* é definida, permitindo que os itens de mídia especi-

Tabela 5.1: Relações Temporais Oferecidas pelo STEVE

Relação Temporal	Símbolo
<i>Equals</i>	
<i>Starts</i>	
<i>Starts With Delay</i>	
<i>Finishes</i>	
<i>Finishes With Delay</i>	
<i>Meets</i>	
<i>Meets With Delay</i>	
<i>Met By</i>	
<i>Met By With Delay</i>	
<i>During</i>	
<i>Overlaps</i>	
<i>Before</i>	



(a) Nenhuma mídia selecionada



(b) Apenas uma mídia selecionada

Figura 5.10: Alertas na criação de uma relação temporal no STEVE

ficados como escravos pelo usuário iniciem suas respectivas apresentações quando a mídia mestre da relação iniciar e terminem quando a mídia mestre finalizar sua apresentação.

Após a definição da relação, o sistema pode exibir setas que representam, graficamente, o elo temporal criado na cadeia entre os itens de mídia participantes da relação. No caso da relação *Equals*, existem duas setas para representar o elo como ilustrado na Figura 5.11. A primeira delas tem sua origem na borda esquerda do retângulo que representa o início da apresentação da mídia mestre da relação. O destino da seta é a borda de início da última mídia escrava em referência a qual linha da cadeia ela ocupa passando

por todas outras escravas da relação. A segunda seta da relação *Equals* tem como origem a borda direita do item de mídia mestre representando o seu término e com destino na última mídia escrava analogamente à representação da primeira seta da relação.

A última escrava de uma relação é definida como a mídia disposta na linha de maior posição, ou seja, a linha mais distante do topo da cadeia temporal, caso as mídias escravas estejam abaixo da mídia mestre do elo na cadeia conforme a Figura 5.11. Quando as mídias escravas estão acima do mestre, o item de mídia denominado como último é o que ocupa a linha mais próxima do topo da cadeia temporal como pode ser visto na Figura 5.11a. Outro cenário exemplificando a configuração da representação da relação *Equals* é mostrado na Figura 5.11b onde existem mídias escravas acima e abaixo do mestre da relação. Este comportamento da representação gráfica da relação referente à presença de mídias escravas acima e/ou abaixo do mestre é aplicado a todas as relações temporais seguintes.



Figura 5.11: Representações gráficas da relação temporal *Equals* no STEVE

A representação gráfica das relações temporais é exibida pelo editor somente quando a opção *Show Relations* estiver selecionada. Esta opção está presente na barra de botões da visão temporal como mostrado na Figura 5.6. Quando esta opção está desmarcada, nenhuma representação será mostrada, apenas os itens de mídia serão dispostos na cadeia temporal conforme o elo temporal escolhido pelo usuário.

É possível notar que a relação *Equals* não está definida na Tabela 4.3 onde as relações temporais do modelo hipermídia proposto são apresentadas. Isto ocorre, pois a relação *Equals* é implementada pela ferramenta utilizando duas relações do modelo hipermídia proposto: *Starts* e *Finishes*, ambas serão analisadas a seguir.

Quanto ao elo temporal *Starts*, o editor proposto possibilita que o autor defina que os

itens de mídia escravos iniciem quando a mídia mestre começa sua apresentação. A disposição das mídias e a representação gráfica desta relação podem ser observadas na Figura 5.12. Esta representação gráfica possui uma seta que tem como origem a borda esquerda do retângulo que representa a mídia mestre. Seu destino é o início da apresentação da última mídia escrava da relação.

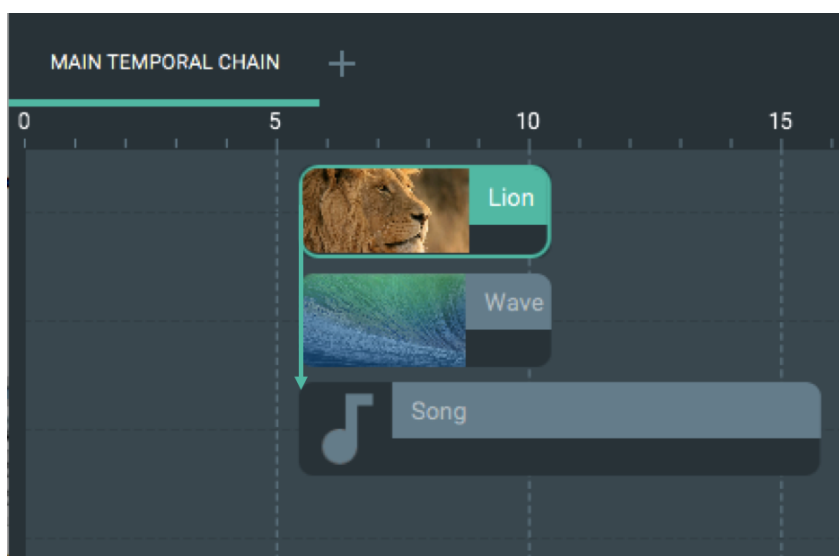


Figura 5.12: Representação da relação *Starts* no STEVE

A relação temporal *Starts With Delay* define que as mídias escravas iniciem com atraso quando a mídia mestre começa. Este atraso deve ser informado pelo usuário assim que ele pressionar a opção dessa relação na barra de botões da visão temporal conforme a Figura 5.13a. Na representação gráfica desta relação temporal, existe uma seta com origem na borda de início da mídia mestre da relação e tendo como destino cada uma das bordas de início das mídias escravas do elo conforme a Figura 5.13b.

Outra relação temporal enunciada na Tabela 5.1 é a *Finishes*. Ela especifica que os nós de mídia escravos devem terminar suas respectivas apresentações quando a mídia mestre finalizar. A Figura 5.14 mostra como este elo é representado graficamente, onde a origem da seta está na borda direita do retângulo que representa a mídia mestre, ou seja, em seu término. O destino da seta está na borda direita da última mídia escrava da relação indicando que os itens escravos e mestre da relação terminam juntos.

Já a relação *Finishes With Delay* estabelece que as mídias escravas terminam  $t$  instantes de tempo após o fim da apresentação do mestre da relação. Como no elo temporal *Starts With Delay*, o usuário também deve especificar o atraso em segundos como na Figura 5.13a. Esta relação é representada por uma seta com início na borda direita do nó mestre do elo e com destino na última mídia escrava da relação passando pelas demais de

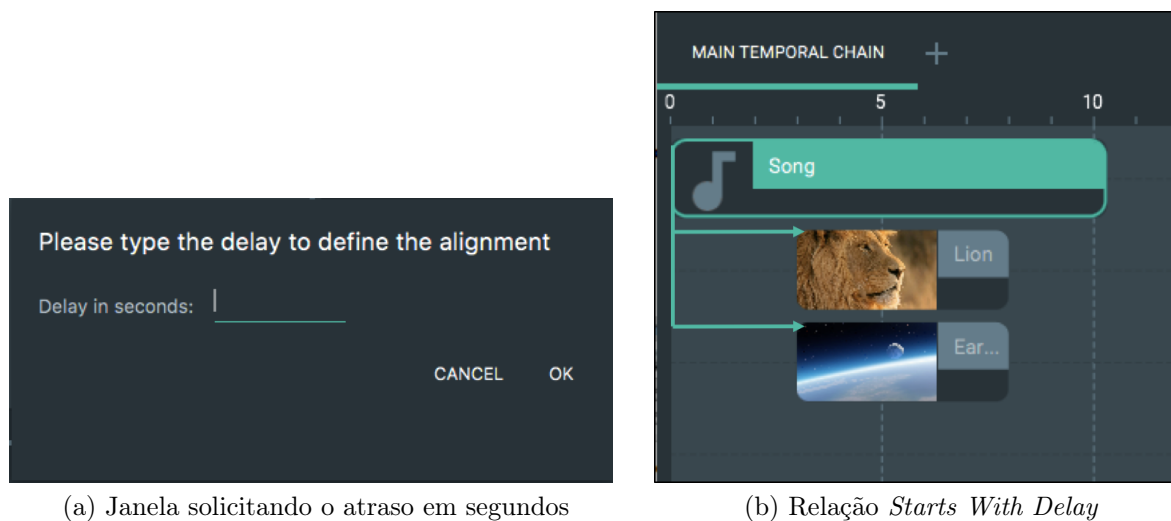


Figura 5.13: Representação da relação Temporal *Starts With Delay* no STEVE

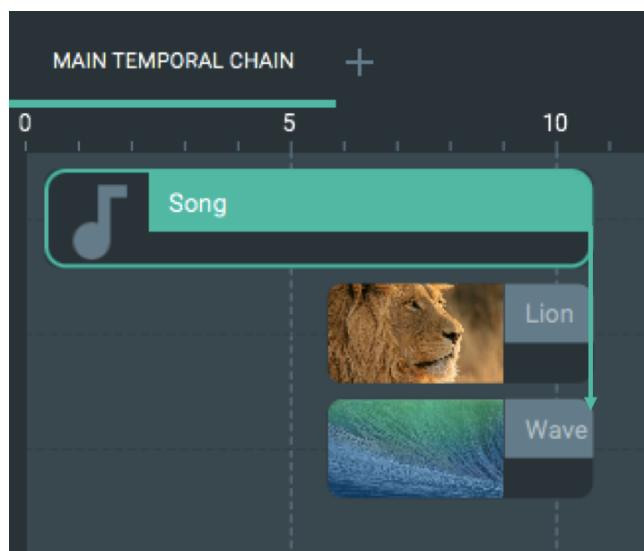


Figura 5.14: Representação da relação *Finishes* no STEVE

acordo com a Figura 5.15.

O elo temporal causal *Meets* determina que as mídias escravas iniciam quando a mídia mestre da relação finaliza. A Figura 5.16 ilustra como as mídias participantes deste tipo de elo se organizam na cadeia temporal e a representação gráfica indicando sua criação no documento hipermídia. Nesta figura, é possível notar que o elo, graficamente, tem a origem da seta no fim da mídia mestre e seu destino na última mídia escrava da relação.

Quanto à relação *Meets With Delay*, ela especifica que os itens de mídia escravos iniciem suas respectivas apresentações após  $t$  instantes de tempo do término da mídia mestre da relação. O atraso  $t$  deve ser especificado pelo usuário através da janela exibida pela ferramenta como mostrado na Figura 5.13a. A representação gráfica desta relação



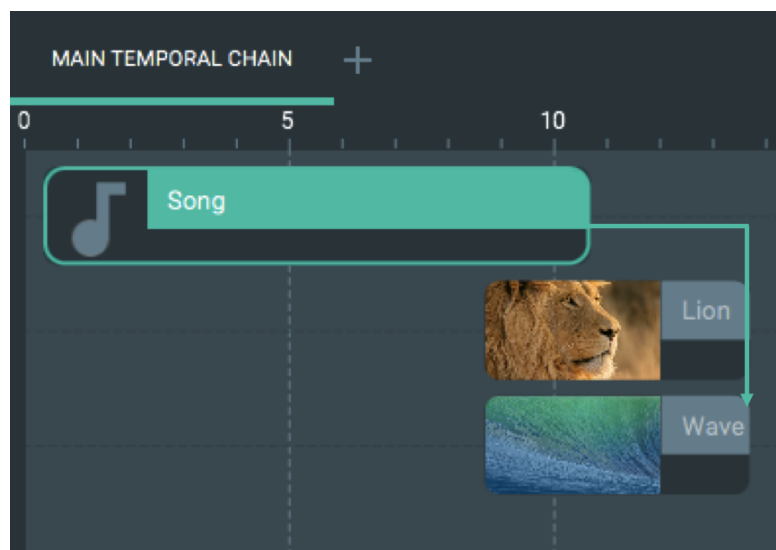


Figura 5.15: Representação da relação *Finishes With Delay* no STEVE

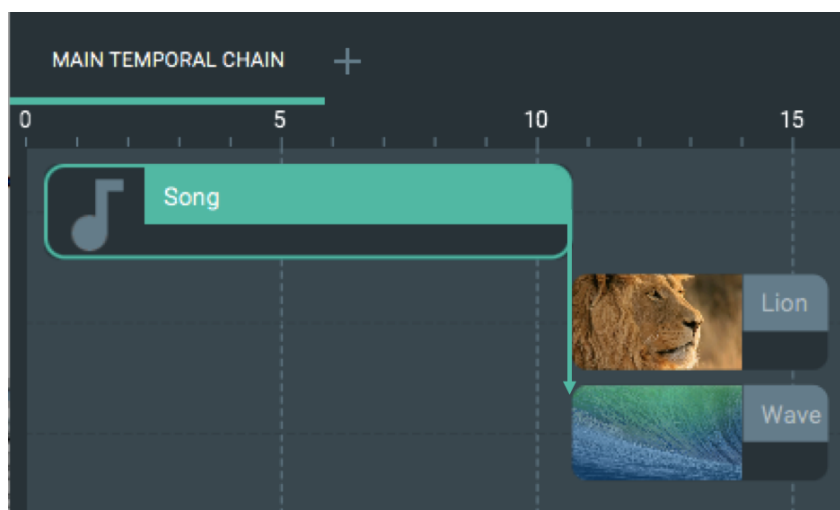
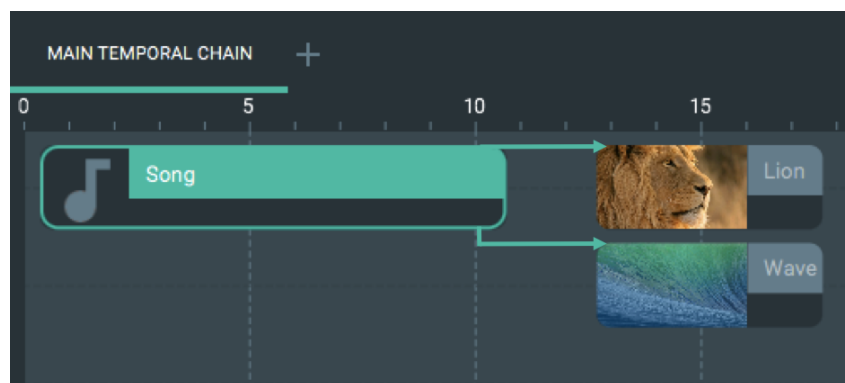
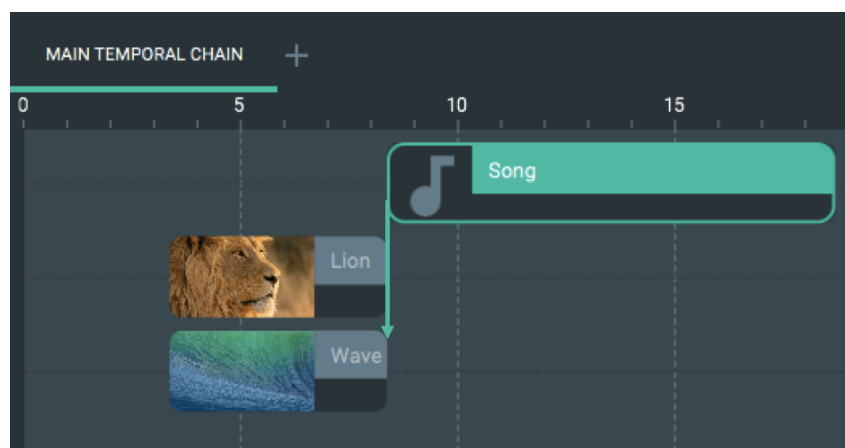


Figura 5.16: Representação da relação *Meets* no STEVE

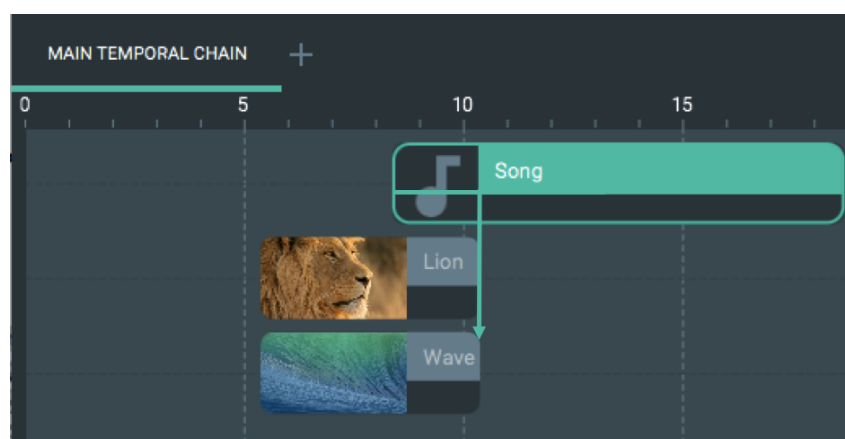
possui uma seta com origem no fim da mídia mestre do elo e o início de cada mídia escrava como destino. A Figura 5.17 ilustra a disposição das mídias participantes de um elo *Meets With Delay* e sua representação gráfica.

A Tabela 5.1 apresenta também a relação *Met By* a qual estabelece que as mídias escravas terminam quando o mestre da relação inicia sua apresentação. A representação gráfica dessa relação tem a origem da seta na borda esquerda da mídia mestre do elo, isto é, em seu início. O destino está no término da última mídia escrava passando pelas demais conforme a Figura 5.18.

A relação *Met By With Delay* define que os itens de mídia escravos da relação devem terminar depois de um determinado atraso após o início da apresentação do mestre da

Figura 5.17: Representação da relação *Meets With Delay* no STEVEFigura 5.18: Representação da relação *Met By* no STEVE

relação. A Figura 5.19 apresenta a indicação gráfica deste elo temporal, onde o início da mídia mestre é a origem da seta e seu destino é a borda da direita da última mídia escrava da relação. Analogamente às demais relações temporais que especificam um atraso para disparar a ação do elo, o editor solicita o valor do atraso para o usuário para definir a relação *Met By With Delay*.

Figura 5.19: Representação da relação *Met By With Delay* no STEVE

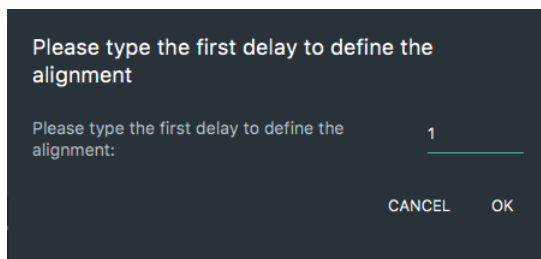
STEVE oferece ainda mais três relações que, semelhantes ao elo *Equals*, não estão presentes no diagrama de classe do modelo SIMM apresentado no Capítulo 4. Elas são implementadas pelo editor utilizando outras relações definidas no modelo proposto.

A primeira delas é a relação *During* a qual especifica que a mídia escrava executa durante a apresentação do mestre. Neste caso, somente um item de mídia pode ser definido como escravo e dois atrasos devem ser especificados pelo usuário. O primeiro deles é referente ao atraso de início da mídia escrava quando a mídia mestre começa sua apresentação, ou seja, a relação *Starts With Delay* é um dos elos utilizados na implementação da relação *During*. Outra relação aplicada na definição do elo *During* é a *Met By With Delay*. Isto torna possível manter uma mesma mídia como mestre tanto nesta relação quanto no elo *Starts With Delay* na criação da relação *During*. Consequentemente, o segundo atraso que deve ser especificado é referente ao intervalo de tempo decorrido do início da mídia mestre.

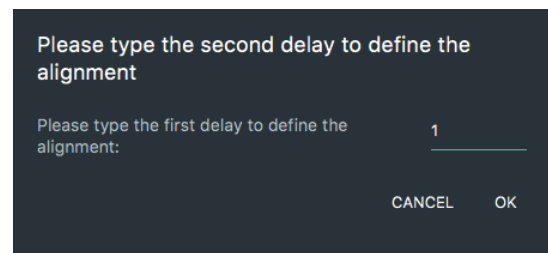
Entretanto, visando facilitar a compreensão do usuário na criação da relação *During*, o segundo atraso que é solicitado pelo sistema é referente ao intervalo contido entre os termos da mídia mestre e escrava respectivamente. A Figura 5.20 mostra as janelas que são apresentadas pelo sistema solicitando os atrasos e a representação gráfica da relação *During*. Ademais, esta figura enfatiza onde os atrasos especificados pelo usuário são aplicados na relação criada.

A segunda relação temporal que não é definida na Tabela 4.3 é a *Overlaps*. Ela estabelece que a mídia escrava deve iniciar depois de um intervalo de tempo quando a mídia mestre começa e deve terminar com atraso quando a mestre finaliza sua apresentação. Portanto, a relação *Overlaps* faz uso das seguintes relações apresentadas no modelo hipermídia proposto: *Starts With Delay* e *Finishes With Delay*. Desta forma, o usuário deve informar ambos os atrasos ao sistema como mostrado nas Figuras 5.20a e 5.20b. Como a relação *During*, o elo *Overlaps* possui somente uma mídia escrava e uma mestre. A representação gráfica desta relação, que pode ser visualizada na Figura 5.21, tem duas setas. A primeira delas possui origem no início da mídia mestre e destino na borda esquerda do nó escravo. A outra seta inicia na borda da direita do mestre da relação e termina o fim da mídia escrava. A Figura 5.21 também indica onde os dois valores de atraso informados pelo autor são aplicados na criação da relação.

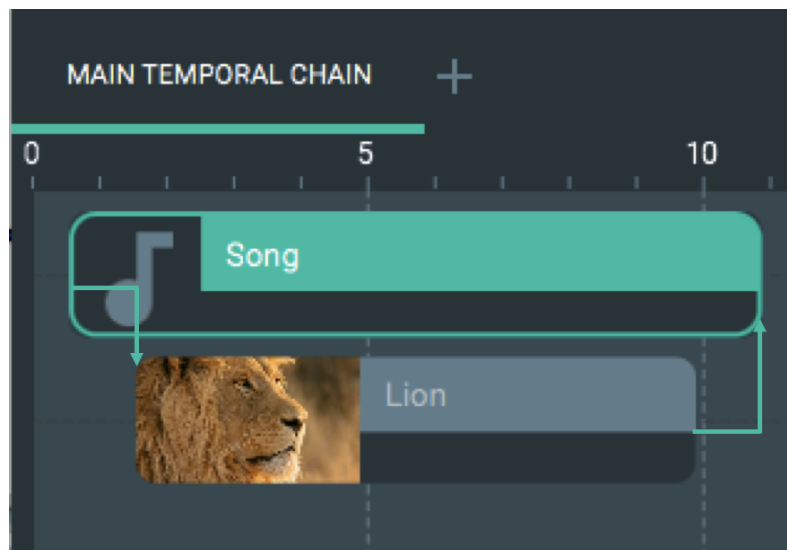
A última relação síncrona que o STEVE oferece ao usuário é o elo *Before*, o qual dispõe as mídias sequencialmente com um intervalo de tempo entre elas. A mídia mestre da relação define o início da sequência de mídias escravas. Desta forma, após  $t$  instantes



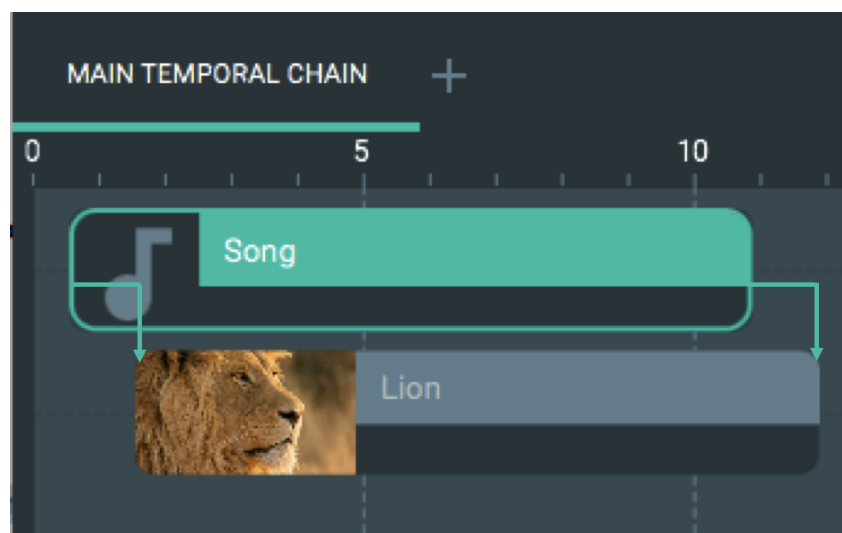
(a) Primeiro atraso



(b) Segundo atraso



(c) Representação gráfica

Figura 5.20: Definido a relação *During* no STEVEFigura 5.21: Representação da relação *Overlaps* no STEVE

de tempo do término da mídia mestre, a primeira mídia escrava inicia sua apresentação. Quando esta termina, o segundo item de mídia escravo inicia também depois do mesmo valor de atraso  $t$  e assim sucessivamente. O valor desse atraso também deve ser informado

pelo usuário como ocorre em outras relações já discutidas. Na Figura 5.22, A representação gráfica da relação *Before* é mostrada. Nela, existem diversas setas, sendo a primeira com origem no fim da mídia mestre e destino no início da mídia escrava seguinte. A segunda seta tem início no fim da primeira mídia escrava e término no início da segunda mídia escrava e assim sucessivamente.

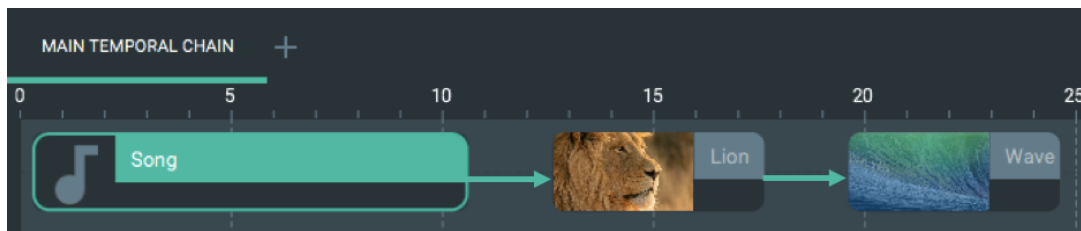


Figura 5.22: Representação da relação *Before* no STEVE

A fim de permitir que o usuário visualize todas as relações já criadas envolvendo uma determinada mídia, ele deve selecioná-la e marcar a opção *Show Relations*. A Figura 5.23 exemplifica esta funcionalidade. Nela, nota-se que o item de mídia *Song* possui diversas relações temporais nas quais tem participação. A primeira delas é a relação *Meets* onde o item *Song* é uma mídia escrava iniciando sua apresentação quando o item de mídia *Lion* termina. Outro elo temporal observado nesta figura, *Starts With Delay*, envolve as mídias *Galaxy* e *Lake* como escravas e o item *Song* como mestre. Desta forma, as mídias escravas iniciam suas respectivas apresentações após 2s do início de *Song*. A relação temporal *Met By* também pode ser vista na Figura 5.23. Ela tem como mestre a mídia *Beach* e o item *Song* participa como escravo. Consequentemente, a mídia *Song* finaliza sua apresentação quando o item *Beach* inicia. A última relação presente é a *Meets With Delay* onde o item de mídia *Song* está associado como mestre e seu término inicia, após 2s, a mídia escrava, *Yosemite*.

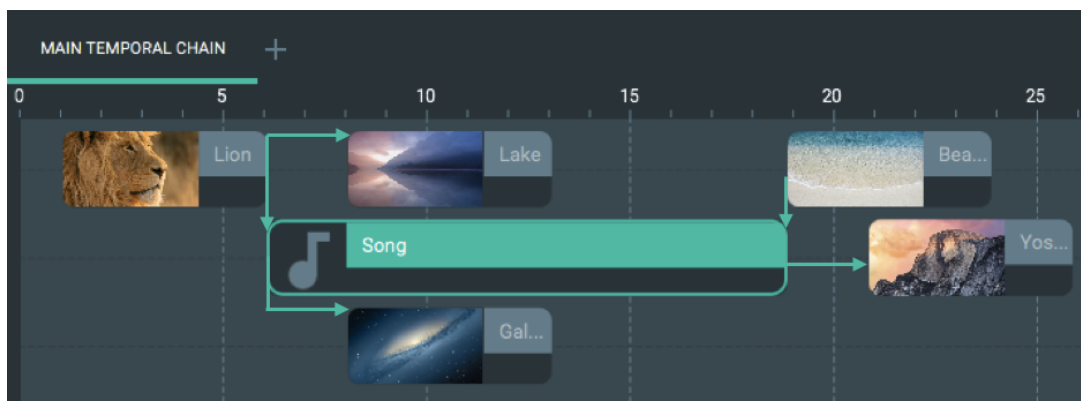


Figura 5.23: Visualizando as relações temporais de uma mídia

Para deletar uma relação temporal já criada entre os itens de mídia em uma cadeia

temporal, o usuário deve realizar os seguintes passos. Como já discutido anteriormente, a fim de exibir a representação gráfica de todas as relações envolvendo uma determinada mídia, o autor necessita, primeiramente, selecionar esta mídia e garantir que a opção *Show Relations*, na barra de botões da visão temporal, esteja marcada. Após isso, o usuário deve selecionar a relação que deseja remover clicando na seta que representa, graficamente, o elo e pressionando a tecla *Delete*. Ou ainda clicar com o botão direito do mouse sobre a seta da relação e selecionar a opção *Delete Relation*.

É importante mencionar que todas as relações temporais discutidas nesta seção possuem condições simples, isto é, apenas uma condição deve ser satisfeita para disparar a ação. Além disso, elas também definem apenas um tipo de ação a ser executado sobre uma ou mais mídias escravas. Isto facilita a criação de relações causais entre os nós de mídia do documento hipermídia, de forma a possibilitar que usuários sem conhecimento em linguagens multimídia padrão possam construir suas aplicações hipermídia utilizando o paradigma baseado em eventos.

### 5.3.3 Suporte à Interatividade

Além de oferecer as relações síncronas discutidas até então para facilitar a sincronização temporal de documentos hipermídia, STEVE também permite que os usuários criem relações de interatividade. Estas permitem que o autor especifique eventos que dependem da interação do espectador com algum nó do documento hipermídia.

A relação de interatividade possui uma mídia mestre, uma lista de mídias escravas e de cadeias temporais conforme discutido no Capítulo 4. Semelhante às relações síncronas, a interatividade possui uma única condição que corresponde ao início do evento de seleção, ou seja, à interação do espectador com a mídia interativa do elo (mídia mestre). Entretanto, ela pode possuir uma ação composta pelo início e término do evento de apresentação de um ou mais itens de mídia. A primeira ação de início de novas mídias está associada somente à lista de cadeias temporais, onde a mídia mestre de cada uma tem sua apresentação iniciada quando o espectador interage, através de algum dispositivo de entrada, por exemplo de um controle remoto, teclado, mouse etc, com a mídia interativa do elo temporal assíncrono. A outra ação é o término da apresentação das mídias escravas, que inclusive a própria mídia interativa pode fazer parte. Em ambas ações, um atraso para disparar cada uma pode ser especificado.

Para definir uma relação de interatividade no STEVE, primeiramente o usuário necessita selecionar algum item de mídia para ser o mestre do elo, isto é, a mídia interativa.

Em seguida, pressionar o botão *Interactivity* presente na barra de ferramentas da visão temporal ou clicar com o botão direito do mouse sobre a mídia e selecionar a opção *Add Interactivity*.

Caso o autor não selecione qualquer mídia em uma cadeia temporal para definir uma mídia interativa através do botão *Interactivity*, o sistema irá exibir uma mensagem de alerta informando que alguma mídia na visão temporal deve ser selecionada para definir uma nova relação de interatividade.

É possível ainda que o usuário adicione mais de uma relação de interatividade para uma mesma mídia, ou seja, dependendo de qual tecla é utilizada para interação com a mídia, o elo dispara as ações correspondentes, estas podendo ser iguais ou distintas. As relações criadas são identificadas com o nome da tecla que dispara o elo. Desta forma, o editor não permite criar duas relações de interatividade que utilizam a mesma tecla interativa.

Por exemplo, considere uma aplicação onde existe um vídeo de início e depois de algum tempo, uma mídia do tipo imagem é exibida. Suponha que a imagem esteja associada a duas relações de interatividade. A primeira elo especifica que, se o usuário interagir com a imagem através do botão verde do controle remoto, a aplicação irá exibir informações sobre o vídeo em execução. Por outro lado, a outra relação define que a aplicação é parada se a seleção da imagem ocorrer com o botão vermelho.

Após selecionar uma mídia e pressionar o botão *Interactivity*, o editor gráfico proposto exibe a janela de criação de relações de interatividade, *New interactive media*, apresentada na Figura 5.24. Nela, três regiões podem ser notadas: *Interactivity Key*, *What Will Be Stopped?* e *What Will Be Started?*.

A primeira delas corresponde aos campos de definição da tecla que deve ser pressionada pelo espectador para interagir com a mídia. O campo *Interactivity key type* se refere ao tipo da tecla e pode assumir os seguintes valores: *Arrows*, *Alphabetical*, *Programming Guide*, *Numeric*, *Channel Change*, *Volume Change*, *Colors* e *Control*. De acordo com a opção do tipo de tecla escolhido pelo usuário, o campo *Interactivity key* apresenta a lista com os valores possíveis para o autor. Desta maneira, quando o valor *Arrows* é selecionado, o campo da tecla exibe uma lista com as seguintes opções: *Leftwards Arrows*, *Rightwards Arrow*, *Upwards Arrow* e *Downwards Arrow* correspondendo às setas para esquerda, direita, cima e baixo respectivamente para ativação do elo interativo conforme a Figura 5.25.

**New interactive media** SAVE

**INTERACTIVITY KEY**

Interactivity key type: Control

Interactivity key: ENTER

**WHAT WILL BE STOPPED?**

☒ Interactive media will be stopped ☐ End application

Media to be stopped:

Delay to stop in seconds: 0

**WHAT WILL BE STARTED?**

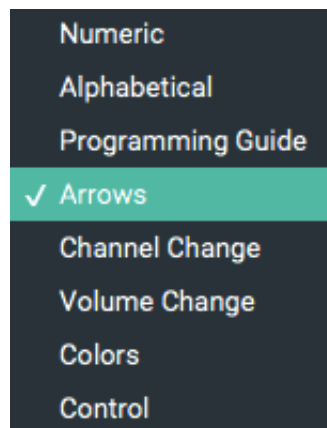
Temporal chain to be started:

Delay to start in seconds: 0

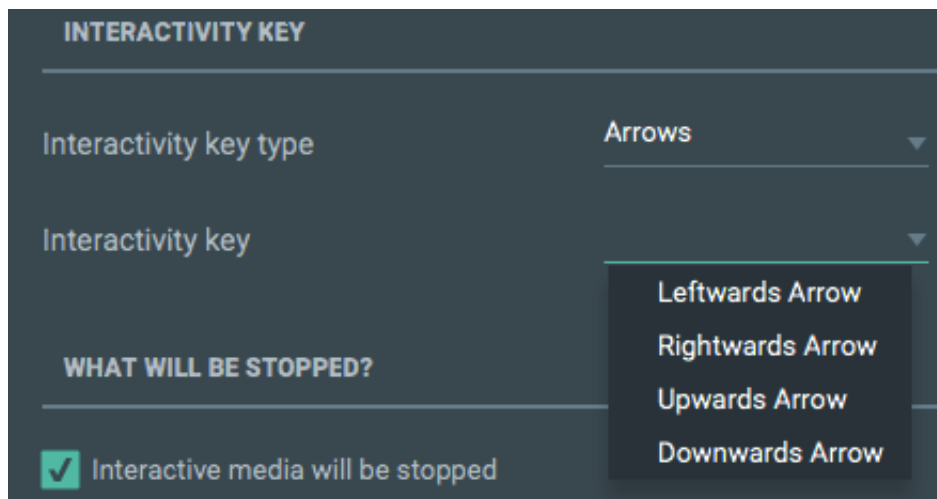
Figura 5.24: Definindo Nova Mídia Interativa

Para a opção *Alphabetical*, uma lista alfabética de A a Z é apresentada. Quando o valor *Programming Guide* é escolhido, o campo da tecla exibe os seguintes botões presentes em um controle remoto de TV: *Menu*, *Info* e *Guide*, teclas aplicadas para os casos em que a aplicação hipermídia criada no STEVE será exportada para a plataforma do Sistema Brasileiro de TV Digital e IPTV padrão ITU. O valor *Numeric* atribui ao campo *Interactivity key* uma lista numérica de 0 a 9 como opções de teclas interativas. Quando o valor *Channel Change* é selecionado, o campo *Interactivity key* exibe as teclas *Channel Up* e *Channel Down* como opções, as quais são adequadas para a plataforma de TV digital. De forma análoga, a opção *Volume Change* atribui os valores *Volume Up* e *Volume Down* ao campo de tecla. Os valores *Red*, *Green*, *Yellow* e *Blue* são dados ao campo *Interactivity key* quando o tipo de tecla *Colors* é selecionado. Elas correspondem às teclas de atalho para acessar conteúdos interativos na TV digital quando disponíveis. A última opção do campo do tipo de tecla é a *Control*. Ela habilita os seguintes valores de tecla para o campo





(a) Opções do campo *Interactivity key type*



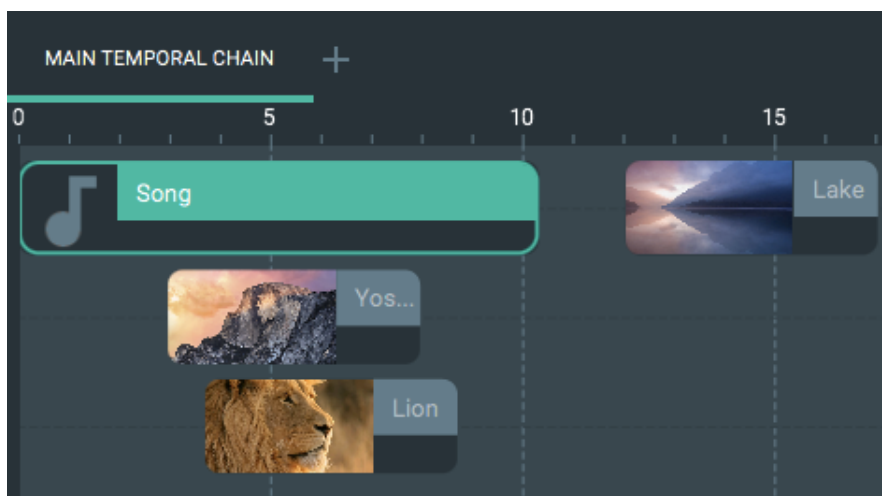
(b) Opções do campo *Interactivity key* para o tipo selecionado

Figura 5.25: Selecionando a tecla interativa

*Interactivity key*: *Back*, *Exit*, *Power*, *Rewind*, *Stop*, *Eject*, *Play*, *Record*, *Pause* e *OK*.

A segunda região, *What Will Be Stopped?*, presente na janela de criação da relação de interatividade apresenta campos para especificar quais itens de mídia terão sua apresentação terminada quando o espectador interagir. O primeiro campo é uma caixa de seleção, *Interactive media will be stopped*, que indica se a própria mídia interativa deve ter sua apresentação terminada quando a interação do espectador ocorrer. Outra caixa de seleção (*End Application*) é exibida ao lado da primeira, para decidir se a aplicação será finalizada quando a interação ocorrer. O terceiro campo, *Media to be stopped*, refere-se às mídias escravas da relação de interatividade que serão paradas após a seleção da mídia interativa. Desta forma, este campo exhibe uma lista de mídias que possui apenas os itens que estão sendo exibidos durante a apresentação da mídia interativa, isto é, ao longo da janela de oportunidade de interação conforme a Figura 5.26. O último campo, *Delay to*

*stop in seconds*, corresponde ao valor de atraso para disparar a ação de término das mídias após a interação do espectador com a mídia interativa do elo.



(a) Mídias sendo apresentadas durante a Janela de oportunidade de interação



(b) Valores disponíveis no campo *Media to be stopped*

Figura 5.26: Selecionando mídias a serem paradas

A última região, *What Will Be Started?*, refere-se à outra ação presente na relação de interatividade que especifica uma lista de cadeias temporais que serão iniciadas com a ativação do elo. Começar uma cadeia significa iniciar a apresentação de sua mídia mestre. Desta forma, o campo *Temporal chain to be started* apresenta todas as cadeias disponíveis na visão temporal. Este campo também permite que o usuário defina uma nova cadeia temporal selecionando a opção *Add new temporal chain*. Caso o autor decida escolher esta última opção, o sistema exibe uma janela para o autor especificar o nome da nova cadeia e a sua mídia mestre.

A primeira informação é representada pelo campo obrigatório *Name* que, por padrão, já é preenchido com o valor *Temporal Chain* mais o número que representa a quantidade de cadeias na visão temporal. O campo *Master Media*, opcional, corresponde ao item de mídia mestre da cadeia que pode ser especificado ou não pelo usuário. Ao salvar a janela de definição da nova cadeia, o sistema verifica se o campo *Name* foi preenchido, caso não esteja, um alerta é mostrado indicando que o usuário deve informar um nome para a nova cadeia.

Além do campo para informar as cadeias temporais a terem suas apresentações iniciadas, a região *What Will Be Started?* também permite especificar um atraso em segundos para iniciar as cadeias no campo *Delay to start in seconds* conforme a Figura 5.24.

A lista de cadeias temporais exibidas no campo *Temporal chain to be started* inclui também a própria cadeia à qual a mídia interativa pertence. Isto permite definir laços (*loops*) na aplicação hipermídia. Por exemplo, considere uma aplicação onde 4 imagens são exibidas sequencialmente com 1s de intervalo entre suas apresentações. Durante toda a exibição da última, uma imagem sobreposta a ela é mostrada representando um botão de voltar a execução do documento. Caso o usuário selecione o botão, a aplicação volta a executar a primeira imagem do documento hipermídia, parando a apresentação da última imagem e do próprio botão de interação.

Para sincronizar esta aplicação temporalmente no STEVE, o usuário deve adicionar 4 imagens na visão temporal e utilizar, por exemplo, a relação *Before* para criar a sequência com todas as imagens, e também inserir o valor 1s para o intervalo de tempo entre as apresentações das mídias na definição da relação. Além disso, o usuário precisa inserir outra imagem na cadeia temporal para representar o botão. Ela será sincronizada com a última imagem da sequência temporalmente utilizando a relação *Equals*. Desta maneira, quando a imagem *Yosemite* inicia, o botão começa sua apresentação e termina no momento que a imagem finaliza. A visão temporal da aplicação criada no STEVE é mostrada na Figura 5.27.

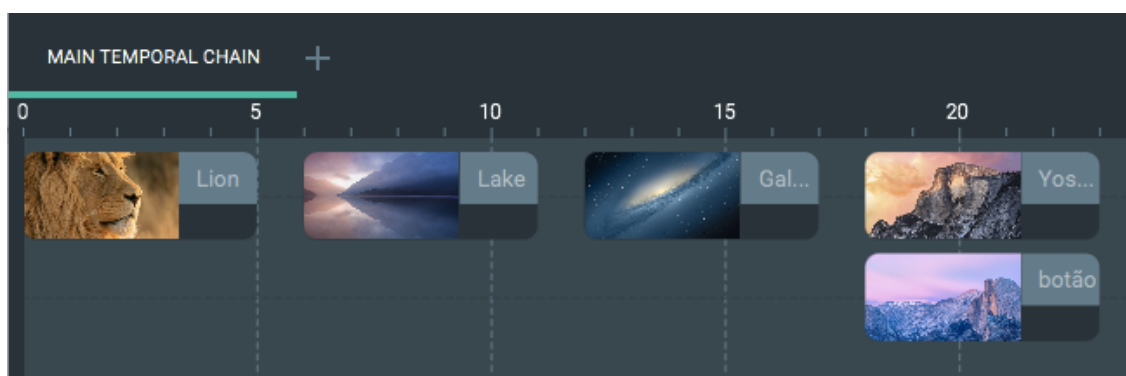


Figura 5.27: Visão temporal da aplicação exemplo

Após isso, o usuário deve especificar o elo de interatividade que será responsável por criar o laço na aplicação. Para isso, o autor deve, primeiramente, selecionar o item de mídia que representa o botão na cadeia temporal e pressionar a opção *Interactivity* na barra de ferramentas da visão temporal. Desta forma, o item que simboliza o botão se tornará uma mídia interativa na aplicação. Em seguida, na janela de definição da relação

de interatividade, o autor deve selecionar a tecla que deve ser pressionada para interagir com a aplicação. Neste exemplo, o botão *Back* foi escolhido. A caixa de seleção *Interactive media will be stopped* foi deixada marcada, visto que o botão terá sua apresentação finalizada. Ademais, a imagem *Yosemite* é inserida no campo *Media to be stopped*, dado que esta também será interrompida. Na região *What Will Be Started?*, o usuário deve adicionar a própria cadeia temporal da mídia interativa, *MAIN TEMPORAL CHAIN*, no campo *Temporal chain to be started* para garantir o laço da aplicação. Nenhum atraso é definido para disparar as ações do elo de interatividade. A janela de definição do elo de interatividade da aplicação hipermídia discutida é exibida na Figura 5.28.

**×** New interactive media SAVE

Interactivity key type Control

Interactivity key Back

**WHAT WILL BE STOPPED?**

☒ Interactive media will be stop... ☐ End application

Media to be stopped Yosemite ×

Add new

Delay to stop in seconds 0

**WHAT WILL BE STARTED?**

Temporal chain to be started MAIN TEMPORAL CHAIN ×

Add new

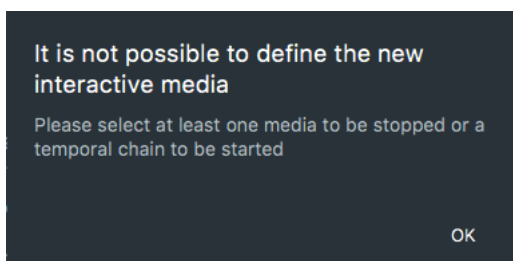
Delay to start in seconds 0

Figura 5.28: Janela de definição da relação de interatividade

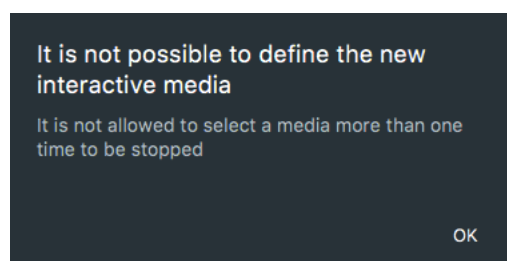
Para os campos *Interactivity key type*, *Interactivity key*, *Interactive media will be stopped*, *Delay to stop in seconds* e *Delay to start in seconds* o editor já define um valor padrão para cada um deles. Assim, quando o usuário seleciona uma mídia para torná-la interativa, a janela de definição da relação de interatividade apresenta tais campos com

os seguintes valores preenchidos por padrão. Para o tipo de tecla, é atribuído o valor *Control* e para a tecla propriamente dita, o valor *OK*. Em relação ao campo *Interactive media will be stopped*, este já é apresentado marcado por padrão, visto que, geralmente, as mídias interativas são paradas com a interação do usuário. E para ambos os campos de atraso, o valor zero é atribuído.

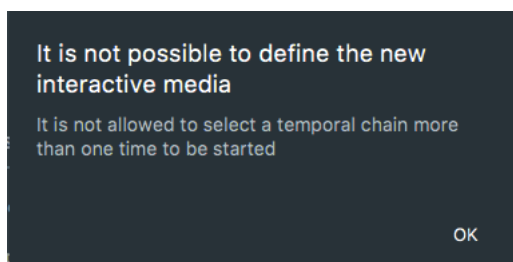
Após o usuário especificar todas as informações necessárias para definir a relação de interatividade, o usuário deve pressionar no botão *Save* que está posicionado na parte superior à direita da janela exibida na Figura 5.28. Ao salvar os dados do elo, STEVE realiza validações dos campos da seguinte maneira. O editor verifica se pelo menos uma mídia foi especificada para ser finalizada ou uma cadeia temporal foi definida para ser iniciada. Desta forma, para satisfazer essa validação o usuário deve marcar o campo *Interactive media will be stopped*, ou adicionar uma mídia ao campo *Media to be stopped*, ou ainda inserir uma cadeia temporal em *Temporal chain to be started*. Outra validação realizada é verificar se existe algum item duplicado na lista de mídias a serem paradas. O sistema também verifica se alguma cadeia temporal está duplicada na lista de cadeias a serem iniciadas. Quando alguma dessas validações falha, o editor proposto exibe alertas informando a inconsistência na definição da mídia interativa como mostra a Figura 5.29. Nela, também pode ser visualizada uma mensagem exibida pelo STEVE quando o usuário tenta fechar a janela de criação do elo de interatividade. Ela questiona se o autor realmente deseja descartar a definição da nova mídia interativa ou não.



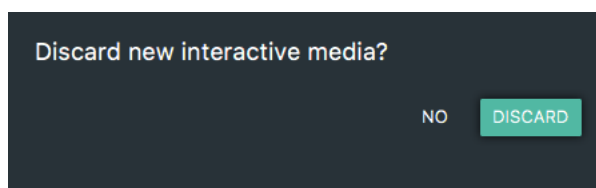
(a) Nenhuma mídia e cadeia temporal inserida



(b) Mídia duplicada



(c) Cadeia temporal duplicada



(d) Mensagem de descarte da relação criada

Figura 5.29: Alertas de validação da relação de interatividade

Tendo passado por todas essas validações, o sistema exibe uma rápida mensagem indicando que a relação de interatividade foi criada com sucesso. Além disso, caso o usuário tenha criado uma nova cadeia temporal durante a definição da mídia interativa, uma nova aba é adicionada na visão temporal representando esta nova cadeia. Ainda após salvar o elo de interatividade, STEVE o reflete graficamente na cadeia temporal. Isto é, como as relações síncronas, o elo de interatividade também possui uma representação gráfica para indicar que a relação foi criada. Ela pode ser visualizada na Figura 5.30.

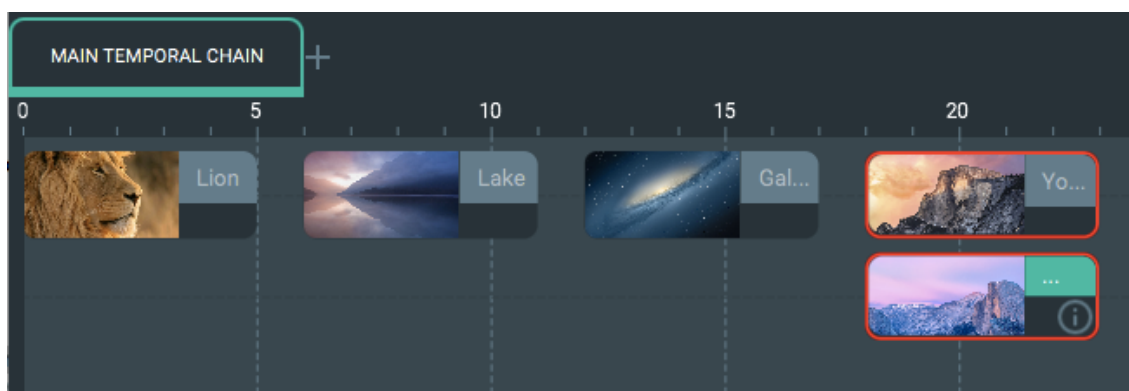


Figura 5.30: Representação gráfica da relação de interatividade

A representação gráfica da relação de interatividade destaca a mídia interativa utilizando um ícone com a letra *i* contida dentro de um círculo conforme visto na Figura 5.30 na única mídia presente na segunda linha da cadeia temporal. As mídias a serem paradas pela relação também são enfatizadas com uma borda vermelha no retângulo que as representa na cadeia temporal. Já as cadeias temporais que devem iniciar suas respectivas apresentações com a ativação do elo são realçadas com uma borda verde em suas abas na cadeia temporal.

Quando o autor deseja visualizar a representação gráfica de qualquer elo de interatividade, basta ele clicar sobre o ícone *i* da mídia ou clicar com o botão direito sobre ela e selecionar a opção *View Interactivity Relations* para exibir a lista de relações de interatividade definidas para a mídia. E, em seguida, escolher a relação desejada.

Para editar uma relação de interatividade, o usuário deve clicar com o botão direito sobre a mídia interativa desejada, selecionar a opção *Edit Interactivity* no menu de contexto e escolher uma das relações listadas. Em seguida, a ferramenta exibe a janela de interatividade já apresentada na Figura 5.28, porém apresentando o título *Edit interactive media*. Todas as informações que definem a relação selecionada para edição são carregadas nos campos e que podem ser editados pelo autor. Após realizar as alterações desejadas, basta o autor confirmá-las clicando no botão *Save* da janela.

Quando o usuário deseja deletar algum elo de interatividade, é necessário que seja aberto o menu de contexto (ou menu popup) da mídia interativa desejada e selecionar a opção *Delete Interactivity*. Logo depois, escolher a relação a ser removida na lista apresentada no menu popup analogamente à edição.

Para simular a interatividade, o usuário deve selecionar a opção *Simulate Interactivity* em *Tools* na barra de menu do STEVE. Logo em seguida, o sistema exibe uma mensagem para verificar se o usuário deseja utilizar o valor padrão para o instante de tempo no qual ocorre a interação do espectador com as mídias interativas da aplicação ou se deseja informar o instante de tempo da ocorrência para cada mídia interativa. Caso o autor escolha esta última opção, STEVE apresenta uma outra janela solicitando o instante para cada mídia interativa. O valor padrão corresponde ao instante de tempo que marca a metade da janela de interação. Ou seja, caso um item interativo seja apresentado a partir do instante 2s a 10s, o valor padrão definido é igual 6s, de forma a dividir a duração (8s) deste item pela metade.

A partir da informação dos instantes de ocorrência dos eventos interativos, a ferramenta gera uma nova cadeia temporal, simulando a ocorrência dos eventos interativos. Esta cadeia resultante pode conter novas mídias e/ou remover outras, conforme a especificação das relações de interatividade. Para ilustrar esta funcionalidade, considere uma aplicação hipermídia que apresenta a cadeia temporal mostrada na Figura 5.31a. Nela, existe um vídeo que inicia a aplicação, após 5s, a imagem *Lion* e *Lake* começam. A primeira termina sua apresentação no instante 15s e a mídia *Lake*, no instante 20s. Neste exemplo, o item *Lion* é interativo e a ocorrência da interação ocasiona o término da própria mídia interativa e do item *Lake*, além de iniciar a nova mídia *Beach*. Dado este cenário, o autor informa ao sistema o instante de ocorrência da interação, por exemplo, 7s, a fim de simular a interatividade. Assim, no instante 7s, os objetos de mídia *Lion* e *Lake* são parados e o item *Beach* é apresentado simultaneamente, de acordo com a Figura 5.31b.

#### 5.3.4 Mantendo a Consistência Temporal

Durante a autoria das apresentações hipermídia no STEVE, o usuário pode realizar edições em seu documento criando relações temporais, arrastando os itens de mídia ao longo da cadeia temporal, deletando mídias da aplicação, alterando a duração delas e removendo cadeias inteiras da visão temporal. Desta forma, a ferramenta possui um tratamento específico para cada uma dessas edições a fim de manter a consistência temporal das aplicações

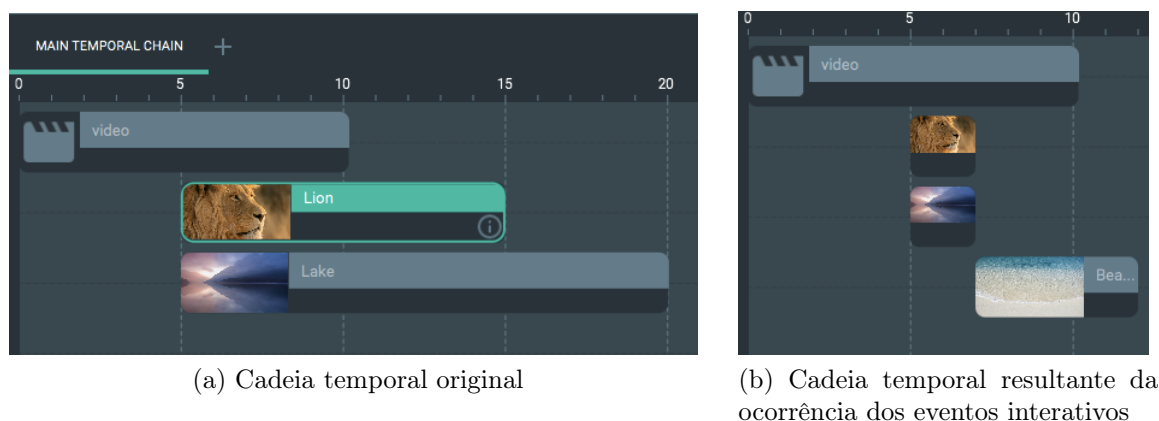


Figura 5.31: Simulação da Interatividade

hipermídia. Cada um desses cenários de edição será discutido nas subseções seguintes. Este tratamento do sistema é fundamental e de suma importância para que os documentos gerados pela ferramenta proposta tenham um comportamento em conformidade com as especificações temporais definidas durante a fase de autoria.

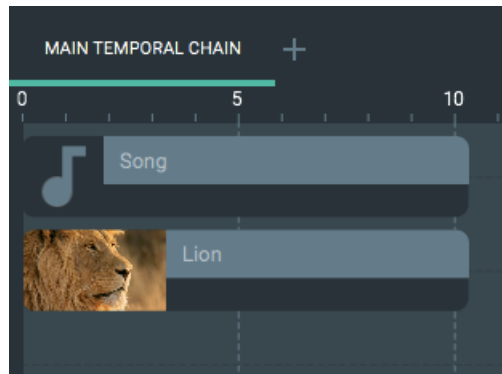
### Criação de Relações Temporais Síncronas

No que diz respeito à criação de relações temporais síncronas no editor proposto, STEVE realiza o seguinte tratamento. Para cada mídia escrava da nova relação temporal criada, o código implementado verifica se esta participa de alguma outra relação, tanto como mestre do elo quanto escrava. Caso a mídia escrava corrente não tenha participado de nenhum outro elo, não existe qualquer conflito, e a nova relação temporal pode ser definida para esta mídia em questão. Por outro lado, quando a mídia escrava faz parte de algum outro elo, o sistema checa, primeiramente, se ela já faz parte de duas relações como escrava. Caso isso seja verdade, o primeiro conflito ocorre para a mídia escrava corrente.

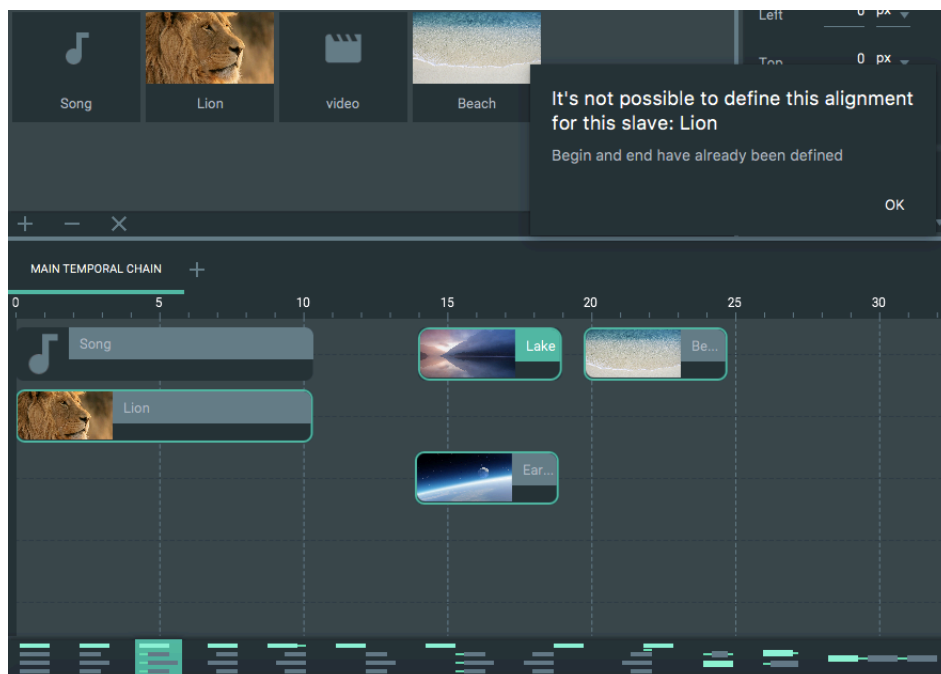
Em outras palavras, isto significa que se uma mídia participa como escrava em duas relações, necessariamente, um dos elos especifica o instante de início da mídia e o segundo define o de término. Por exemplo, suponha-se que o usuário, durante a criação de sua aplicação hipermídia, crie uma relação temporal do tipo *Equals* entre dois itens de mídia conforme a Figura 5.32a. Lembre-se que esta relação está implementada pelo modelo hipermídia proposto fazendo o uso das relações *Starts* e *Finishes* conforme visto no Capítulo 4. Após essa definição, o autor tenta criar outra relação temporal, o elo *Start With Delay*, cujo mestre é o item de mídia *Lake*. O item de mídia *Lion*, que já é escravo da relação *Equals*, é especificado como uma das mídias escravas do elo como ilustrado na Figura 5.32b. Logo depois da tentativa, o editor exibe um alerta, conforme a Figura



5.32b, informando que não é possível definir a relação temporal para a mídia escrava *Lion*, pois seu instante de início e fim já estão definidos.



(a) Relação existente: *Equals*



(b) Alerta do conflito ao definir a Relação *Start With Delay*

Figura 5.32: Conflito: início e fim já definidos

É importante enfatizar que as validações realizadas na criação das relações são feitas para cada um de seus itens escravos selecionados na cadeia temporal. Assim, no exemplo anterior, para as outras mídias selecionadas como escravas, é possível definir a relação temporal *Start With Delay*, visto que elas não participam de nenhuma relação. A Figura 5.33 representa o resultado da criação dessa relação mostrando a nova posição no tempo das mídias escravas, para as quais o elo foi possível especificar.

Ao final da validação de todos os itens selecionados como escravos para a relação temporal que está sendo criada, o sistema verifica se foi possível definir o elo para algum desses itens. Caso não tenha sido possível criar o elo, o editor exibe uma mensagem infor-

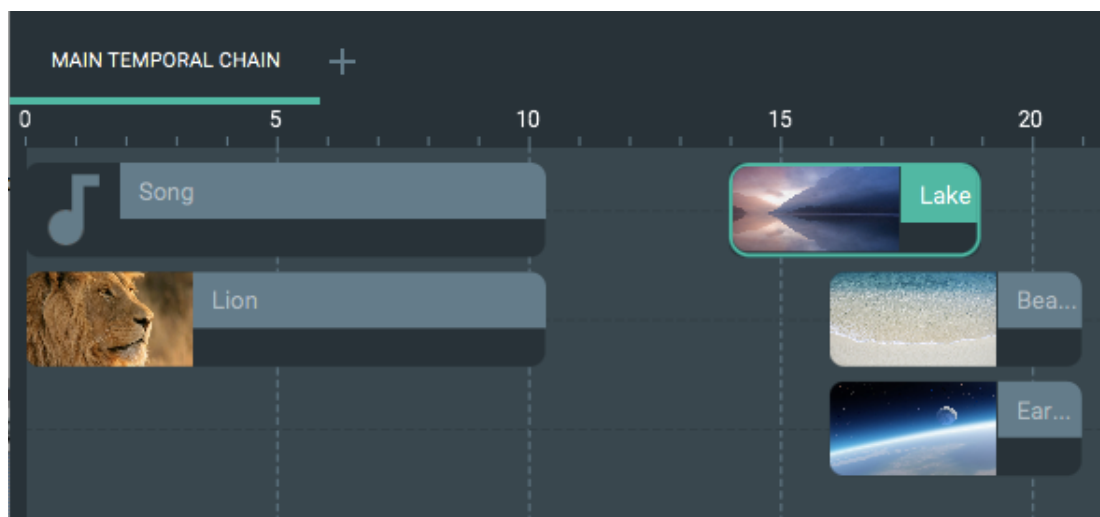


Figura 5.33: Resultado da criação do elo *Starts With Delay*

mando ao usuário que nenhuma mídia selecionada pode ser escrava da relação conforme a Figura 5.34.

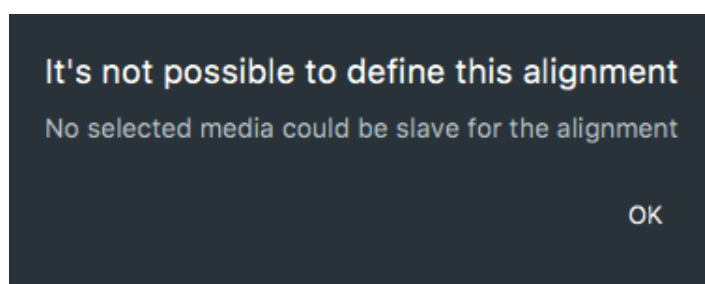
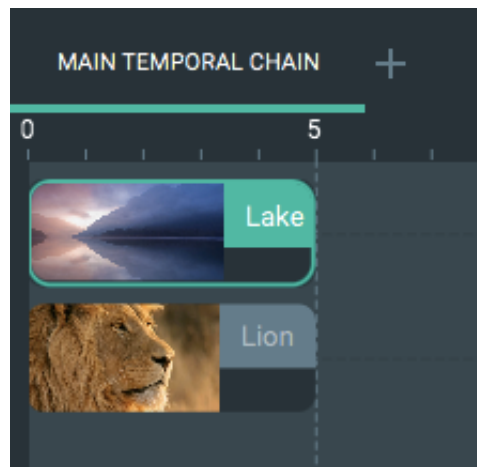


Figura 5.34: Nenhuma mídia selecionada pode ser escrava da relação

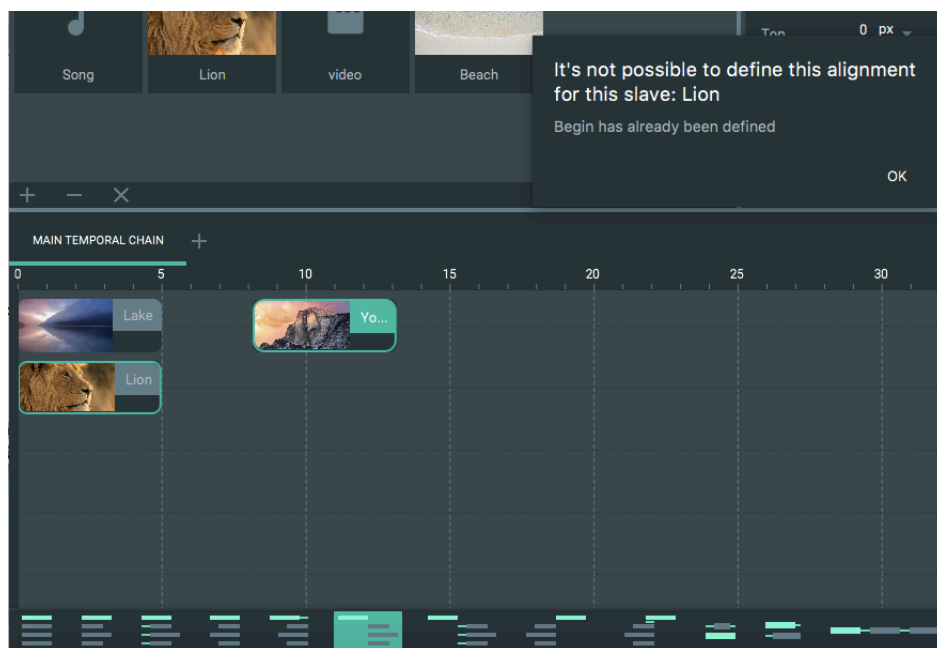
O primeiro conflito descrito ocorre quando a mídia escrava corrente (item da lista de mídias selecionadas como escravas para a nova relação) já participa de duas relações como escrava. Para os casos onde a mídia corrente já participa somente de uma relação, o sistema implementa o seguinte tratamento. Primeiramente, é verificado se a nova relação define o mesmo que o elo ao qual a mídia corrente pertence, ou seja, se ambas as relações especificam um instante de início para a mídia escrava ou um instante de término.

Para exemplificar o conflito, suponha que o usuário tenha criado uma relação do tipo *Starts* envolvendo os itens de mídia *Lake*, mestre do elo, e *Lion* como a mídia escrava da relação. A configuração no tempo dessas mídias é ilustrada na Figura 5.35a. Em seguida, o autor decide criar outra relação temporal, *Meets*, tendo como um dos seus escravos, o item *Lion* conforme destacado na Figura 5.35. Esta nova relação também especifica um instante de início para suas mídias escravas. Desta maneira, o sistema mostra uma mensagem informando ao usuário que não é possível definir a relação desejada, pois o

instante de início da apresentação do item de mídia *Lion* já está especificado, neste caso, pela relação *Starts*.



(a) Relação existente: *Starts*

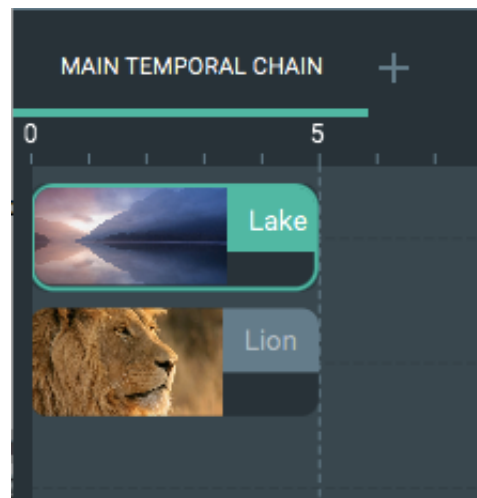


(b) Alerta do conflito ao definir a relação *Meets*

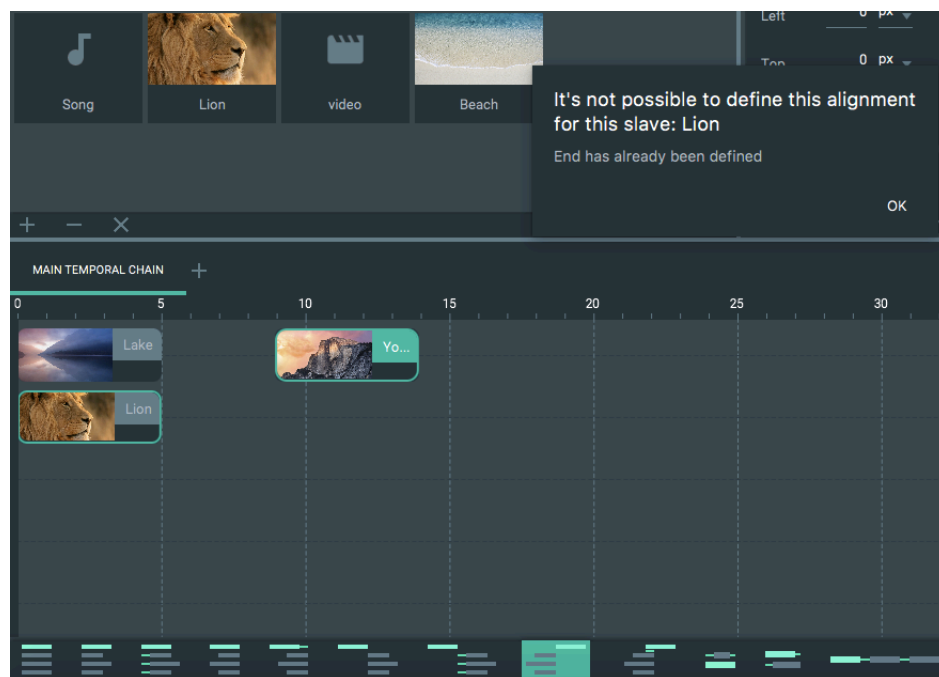
Figura 5.35: Conflito: início já definido

Continuando a análise sobre o caso no qual existe uma relação que define o mesmo que o novo elo, o exemplo apresentado na Figura 5.36a mostra o cenário onde o instante de fim de uma mídia é definido pela relação *Finishes*. Desta forma, a apresentação da mídia *Lion* termina com o fim da mídia *Lake*, mestre da relação. Dado este cenário, o autor deseja criar uma relação do tipo *Met By* envolvendo a mídia *Lion* como escrava de acordo com a Figura 5.36b. Uma vez que esta relação também define o fim de suas mídias escravas, o sistema alerta o usuário de um conflito, conforme a Figura 5.36b, informando que o instante de término para a mídia *Lion* já está estabelecido. Neste exemplo, pela

relação *Finishes*.



(a) Relação existente: *Finishes*



(b) Alerta do conflito ao definir a relação *Met By*

Figura 5.36: Conflito: término já definido

Ainda em relação ao caso onde a mídia escrava corrente do elo a ser criado participa apenas de uma outra relação, os seguintes conflitos também podem ocorrer. O primeiro deles é quando o instante de início definido pela relação a ser criada é maior que o instante de término por um outro elo. A fim de elucidar este conflito, considere que a relação *Finishes* esteja definida envolvendo as duas mídias *Lake* e *Lion* como mestre e escrava respectivamente conforme a Figura 5.36a. Dado este cenário, o autor tenta criar uma relação do tipo *Starts* tendo o item de mídia *Beach* como mestre e *Lion* como escravo. Esta configuração pode ser visualizada na Figura 5.37. Visto que o item *Beach* está após

ao item *Lake* na cadeia temporal, o instante de início definido pela a relação *Starts* é maior que o instante de término do elo *Finishes*. Assim, o sistema exibe um alerta, mostrado na Figura 5.37, informando ao usuário que não é possível especificar a relação, pois o início definido por ela é maior que o fim estabelecido por uma outra relação para a mídia escrava corrente, neste exemplo, o item *Lion*.

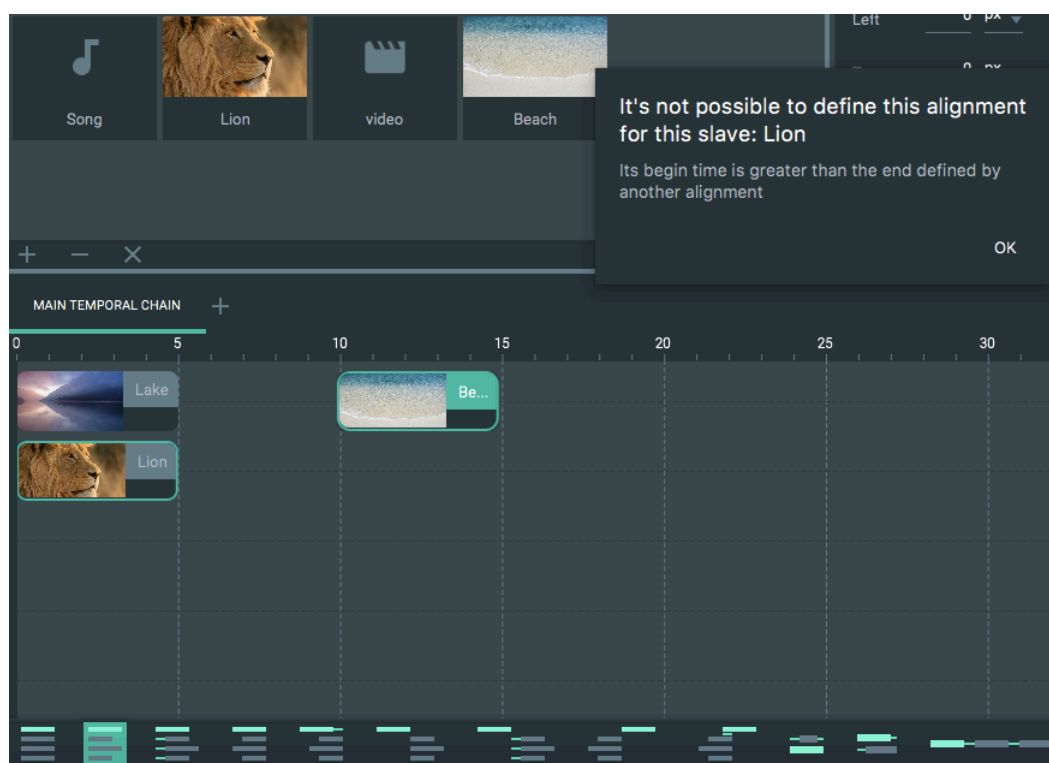


Figura 5.37: Conflito: Novo início é maior que o fim definido por outra relação

O segundo conflito, para o cenário no qual a mídia escrava corrente da relação a ser definida participa apenas de uma outra relação, ocorre quando o instante de término especificado pela nova relação é menor do que o instante de início definido por um outro elo. Exemplificando, tome uma relação do tipo *Meets* definida para as mídias *Lake* e *Lion*, as quais participam como mestre e escrava respectivamente, como ilustrado na Figura 5.38a. Em seguida, o autor tenta criar uma segunda relação do tipo *Finishes* tendo como mídia escrava, o item *Lion*. A mídia mestre desta relação, *Beach*, termina antes do início do item *Lake*, mestre da relação *Meets*. Com este cenário, o STEVE mostra um alerta logo após a tentativa de definição da nova relação de acordo com a Figura 5.38b. Esta mensagem visa informar ao usuário que não foi possível estabelecer a relação desejada, pois o instante de término especificado para a mídia escrava corrente da nova relação é menor do que seu instante de início definido por um outro elo.

Mesmo quando uma mídia escrava já tiver seu instante de início estabelecido por um

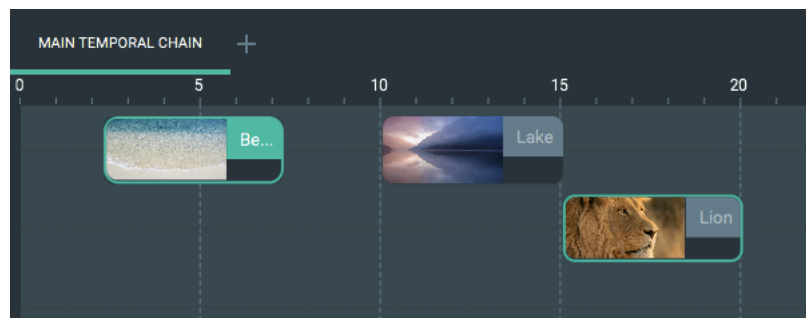
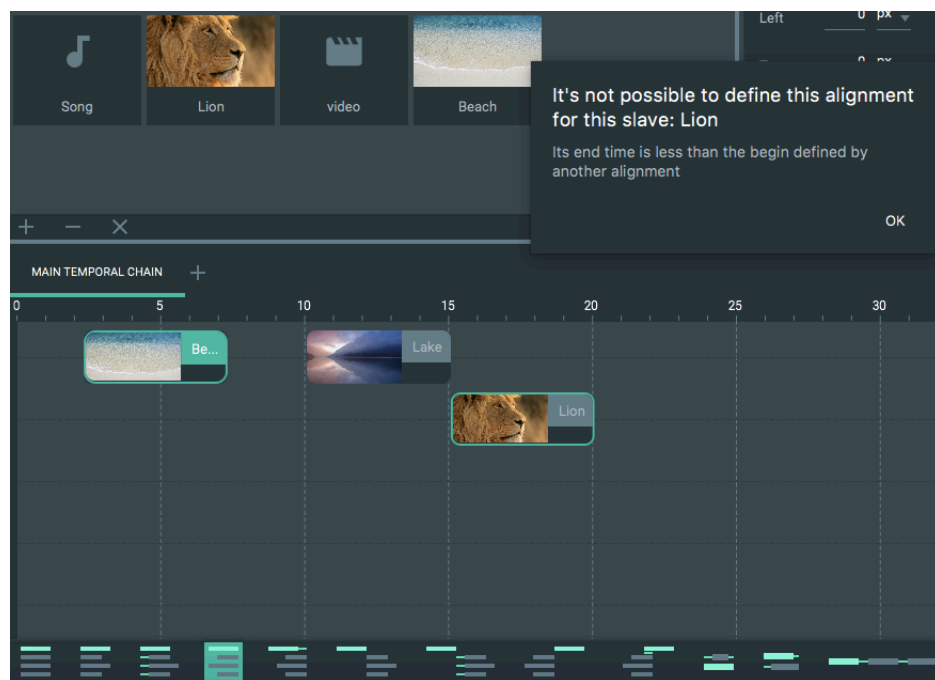
(a) Relação existente: *Meets*(b) Alerta do conflito ao definir a relação *Finishes*

Figura 5.38: Conflito: Novo término é menor que o início definido por outra relação

elo temporal, uma outra relação pode especificar, normalmente, o seu instante de término desde que o novo fim não seja menor que seu instante de início. Neste caso, o sistema altera a duração da mídia de forma a satisfazer ambas as relações temporais. Como exemplo, considere uma aplicação multimídia, na qual a mídia *Lake* é o mestre da relação *Starts* definindo, assim, o instante de início do item escravo, *Lion*, como mostrado na Figura 5.35a. A partir deste cenário, o usuário cria um outro elo do tipo *Finishes* tendo como a mídia escrava, o item *Lion*, e a mídia *Beach* fazendo o papel de mestre. Como pode ser observado na Figura 5.39, o mestre desta relação inicia sua apresentação alguns segundos depois do fim da mídia *Lake*. Desta forma, para definir a relação *Finishes* mantendo o elo *Starts*, STEVE altera a duração da mídia *Lion* aumentando o seu instante de término. O resultado dessas duas relações aplicadas à mídia *Lion* pode ser visualizado na Figura 5.39.

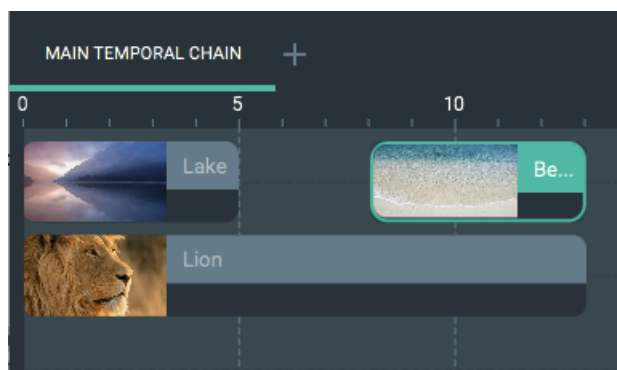
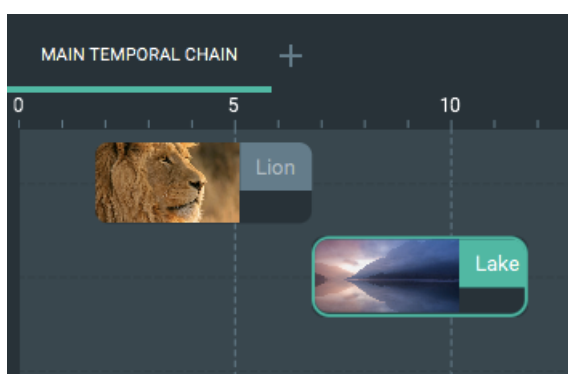
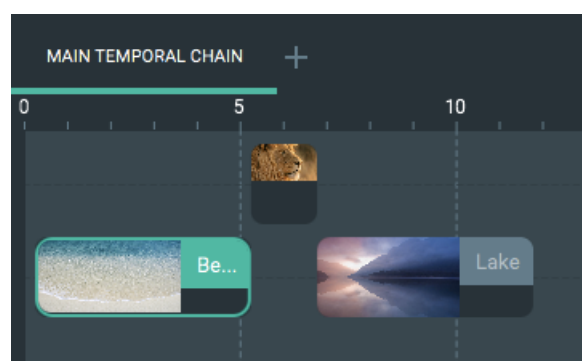


Figura 5.39: Definindo uma relação alterando o instante de término da mídia

Analogamente, ainda que o instante de término de um item de mídia escravo já se encontre definido por uma relação temporal, o usuário pode criar um outro elo que estabeleça seu instante de início desde que ele não seja maior que seu fim. Nessa situação, a ferramenta modifica a duração da mídia escrava visando atender às duas relações, uma que define o fim da mídia escrava e outra, o início. Para ilustrar este cenário, suponha que a relação temporal *Met By* esteja definida da seguinte maneira. O mestre é representado pela mídia *Lake* e sua escrava é a mídia *Lion* conforme a Figura 5.40a. Nesta circunstância, o autor deseja construir uma relação do tipo *Meets* com a mídia *Beach* sendo o mestre e o item *Lion* como escravo. Note que neste exemplo, o fim da mídia *Beach* ocorre antes do início do item *Lake*. Assim, ao definir a relação *Meets*, o editor altera o instante de início da mídia *Lion*, neste caso, diminuindo a duração deste item como pode ser notado na Figura 5.40b.



(a) Cenário exemplo possuindo a relação *Met By*



(b) Resultado das duas relações *Met By* e *Meets*

Figura 5.40: Definindo uma relação alterando o instante de início da mídia

Após verificar os conflitos que podem ocorrer quando a mídia escrava corrente do novo elo ser definido está presente também como escrava em outras relações (no máximo duas), o sistema checka se existe alguma relação, na qual a mídia corrente atua como mestre. Caso seja verdade, o sistema arrasta todos os itens de mídia associados a ela

direta ou indiretamente. Este processo de arrastar mídias dependentes quando uma é arrastada será visto em detalhes mais adiante.

A fim de resumir os conflitos ocorridos nesta validação, a Tabela 5.2 apresenta uma descrição para cada um desses conflitos e o cenário onde eles acontecem.

Tabela 5.2: Conflitos na Criação das Relações Temporais Síncronas

Descrição do Conflito	Cenário Onde Ocorre
Início já definido	Nova relação tenta definir o início de uma mídia que já possui seu começo especificado por uma outra relação
Término já definido	Nova relação tenta definir o término de uma mídia que já possui seu fim especificado por uma outra relação
Início e término já definidos	Tentativa de criar qualquer relação para uma mídia que já possui seu início e término definidos por outras relações
Novo início é maior do que o fim existente	Nova relação tenta especificar um início para uma mídia cujo término é menor que este início
Novo fim é menor do que o início existente	Nova relação tenta especificar um término para uma mídia cujo início é maior que este fim

### Reposicionamento de Mídias

Quando o usuário deseja alterar o período de apresentação de um item de mídia, ele deve arrastar a mídia ao longo da cadeia temporal conforme apresentado na Seção 5.3.1. Nessa situação, o sistema também realiza um tratamento visando a consistência temporal do documento que está sendo editado no STEVE. Assim, no momento em que o autor seleciona e arrasta um item de mídia na cadeia temporal, o editor também arrasta, automaticamente, todos os objetos de mídia que estão ligados direta ou indiretamente ao item arrastado pelo usuário.

Um item de mídia está ligado diretamente a outro quando ambos participam de uma mesma relação temporal, seja como escravo ou mestre. Ainda que ambos os itens de mídia não façam parte de uma mesma relação, eles podem estar ligados entre si, neste caso, indiretamente, ou seja, a relação temporal direta de um item *X* com *Y* e do objeto de mídia *Y* com outro *Z* acarreta uma associação indireta entre os itens de mídia *X* e *Z*.

Para ilustrar este tratamento dado pelo sistema durante o reposicionamento de itens na cadeia temporal, suponha a configuração da cadeia mostrada na Figura 5.41a. Nela, existem quatro relações temporais: *Meets*, *Meets With Delay*, *Starts With Delay* e *Starts*,



as quais foram indicadas por  $R1$ ,  $R2$ ,  $R3$  e  $R4$  respectivamente. A partir deste cenário, considere que o usuário deseja arrastar o item de mídia *Lion*, destacado na Figura 5.41a e pertencente à relação  $R1$ . É possível notar que objeto *Lion* está ligado ao *Lake* diretamente através da relação  $R2$ . E que o item *Lake* está associado diretamente aos objetos de mídia *Beach* e *Wave*, escravos da relação  $R3$ . Diante deste cenário, quando o usuário arrasta o item de mídia *Lion* para uma nova posição na cadeia temporal, todos os itens pertencentes aos elos  $R1$ ,  $R2$  e  $R3$  também são levados juntos a esta nova posição, sem perder a sincronização temporal definida por estas relações conforme a Figura 5.41b. Isto é, nenhuma relação é deletada ao arrastar os itens de mídia. Ainda pode ser observado nesta última figura que os itens pertencentes à relação  $R4$  permaneceram na mesma posição quando *Lion* foi arrastado, pois nenhum deles está ligado a qualquer objeto de mídia das demais relações.

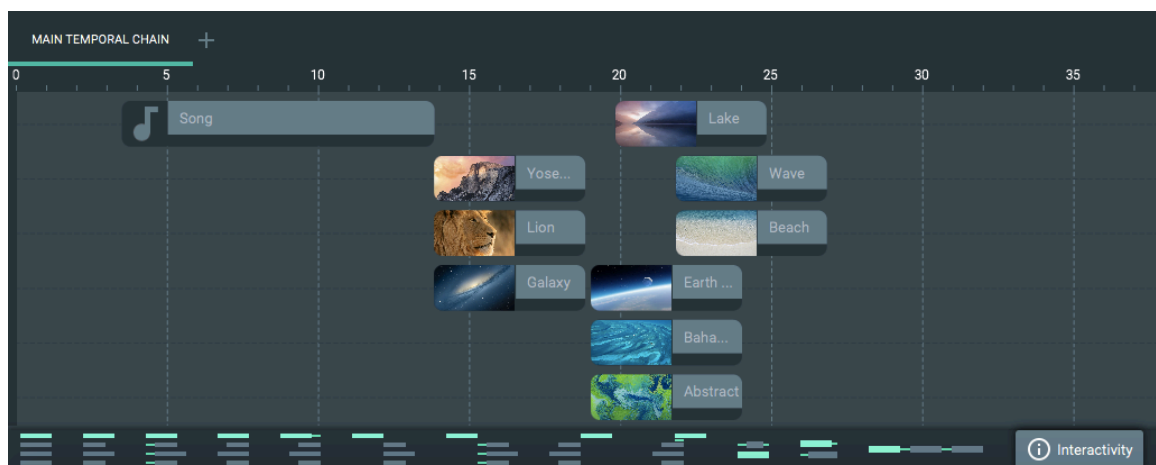
(a) Cenário com as relações  $R1$ ,  $R2$ ,  $R3$  e  $R4$ (b) Configuração da cadeia temporal após arrastar a mídia *Lion*

Figura 5.41: Comportamento do STEVE quando uma mídia é arrastada

Representando a visão temporal na teoria dos grafos, os itens de mídia, as relações

e as cadeias temporais podem ser simbolizados pelos vértices, pelas arestas e por grafos respectivamente. Desta maneira, um conjunto de objetos de mídia associados direta ou indiretamente representa um componente conexo, ou seja, um subgrafo maximal conexo de um grafo  $G$ , a cadeia temporal. No exemplo apresentado na Figura 5.41, a cadeia temporal representa um grafo desconexo onde existem dois componentes conexos. O primeiro é representado pelo conjunto de objetos de mídia e de relações arrastados na operação do usuário como mostrado na Figura 5.41b. O segundo é simbolizado pelo conjunto formado pelos itens participantes de  $R_4$  e pela própria relação. Assim, quando o autor arrasta algum item de mídia na cadeia temporal, somente os objetos de mídia pertencentes ao seu componente conexo são arrastados também mantendo a sincronização temporal especificada pelas relações temporais existentes. Desta forma, STEVE implementa esta funcionalidade através de uma busca em profundidade para percorrer todos os itens de mídia do componente conexo ao qual o objeto arrastado pertence.

Quando um item de mídia interativo é arrastado pelo autor ou automaticamente pelo sistema, o editor se comporta de forma análoga ao que foi descrito no exemplo da Figura 5.41. Além disso, o sistema verifica se os objetos de mídia a serem finalizados, conforme a definição da relação de interatividade, ainda estão sendo apresentados durante o novo período de apresentação do mestre do elo interativo. Para os itens que não estão mais sendo apresentados, STEVE os remove da lista de itens a serem parados, campo apresentado na janela de definição de interatividade.

### Removendo Mídias e Cadeias Temporais

O sistema também implementa uma lógica de verificação das relações quando o usuário remove um item de mídia ou uma cadeia temporal. No que diz respeito à remoção de mídias, o editor proposto verifica, primeiramente, se existe alguma relação temporal, síncrona ou de interatividade, que fazia referência ao item deletado. Caso isto se confirme, o sistema checa se o objeto removido participava como mestre ou escravo da relação. Se o item for mestre, STEVE remove automaticamente a relação temporal. Quando a mídia deletada é escrava, no caso de relações síncronas, o sistema remove o elo somente se a lista de mídias escravas da relação torna-se vazia após o autor ter removido o item de mídia, visto que o elo não terá mais efeito sobre nenhum evento de apresentação de qualquer item de mídia da visão temporal. Para relações de interatividade, o elo é removido quando não só a lista de mídias escravas torna-se vazia como também a lista de cadeias a serem iniciadas devido à interação do espectador.

Além disso, quando o item removido é a mídia mestre de uma cadeia temporal, o sistema deve definir um novo mestre. Para isso, STEVE seleciona o item de mídia com menor instante de início como a nova mídia mestre da cadeia. Ademais, os usuários ainda podem deletar cadeias temporais, o que também deve ser tratado pelo sistema a fim de manter a consistência temporal. Desta maneira, quando uma cadeia é removida, o sistema verifica se esta fazia parte de alguma relação de interatividade. Caso isto não se confirme, o sistema remove apenas a cadeia e não realiza nenhuma modificação nas relações temporais do documento. Do contrário, o editor checa se a lista de cadeias a serem iniciadas deste elo de interatividade se tornou vazia. Se de fato ela não possuir mais elementos e a lista de mídias a serem paradas da relação de interatividade estiver também vazia, a ferramenta remove este elo.

### Alteração da Duração

Outra funcionalidade oferecida pelo sistema que também requer um tratamento específico para manter a consistência temporal das apresentações hipermídia é a alteração da duração dos itens de mídias. Para o usuário aumentar ou diminuir a duração, ele pode alterar o instante de início ou fim da mídia. Assim, o sistema verifica se estes instantes já estão definidos por alguma relação temporal, onde o item selecionado para ter sua duração modificada participa como escravo. Caso algum deles já tenha sido especificado, o editor bloqueia a alteração do usuário. Por exemplo, considere uma aplicação multimídia, onde existe uma relação do tipo *Start With Delay* que define o instante de início das mídias *Lion* e *Lake*. A partir desta configuração, caso o autor deseje alterar a duração do item *Lion* arrastando a borda lateral esquerda, a qual representa o início da mídia, o sistema exibirá uma mensagem de alerta bloqueando esta ação.

Outro cenário que pode ocorrer durante a edição da duração é quando o item modificado é mestre de alguma relação temporal. Neste caso, o sistema atualiza a posição das mídias escravas conforme o tipo da relação. Ou seja, se o elo define o instante de início dos itens escravos e o começo do mestre é alterado, o sistema atualiza o instante de início de cada um de forma a manter a relação causal existente entre o mestre e seus escravos. Analogamente, se a relação define o instante de término dos escravos e o fim do mestre é alterado, o editor modifica o instante de fim de cada um a fim de preservar o relacionamento temporal entre os itens de mídia.

Para ilustrar essa situação, suponha a cadeia temporal mostrada na Figura 5.42a onde a relação temporal *Meets* é definida para o item *Lion*, mestre do elo, e para os objetos de

mídia *Lake* e *Beach*. Assim, quando o instante de fim de *Lion* é aumentado para alterar a duração da mídia, automaticamente, o sistema atualiza o instante de início de cada escravo com o valor do instante de término da mídia mestre para manter a relação *Meets* consistente como mostrado na Figura 5.42b.

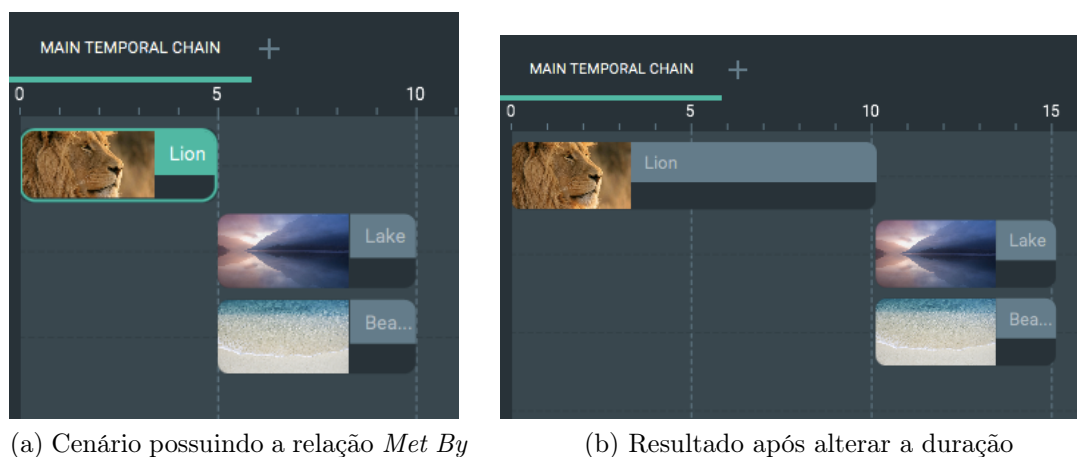


Figura 5.42: Comportamento do STEVE quando a duração de uma mídia mestre é alterada

## 5.4 Visão Espacial

Além de oferecer a edição da visão temporal das aplicações hipermídia, STEVE proporciona a edição da visão espacial, permitindo aos usuários especificarem onde e como os itens de mídia irão se apresentar durante toda a execução do documento. Para isso, a ferramenta proposta apresenta painéis de propriedades que definem a posição, tamanho e estilo de um item de mídia. O editor proposto ainda exibe uma região que permite visualizar a configuração espacial e de aparência das mídias apresentadas num determinado instante de tempo, realizando uma prévia da apresentação do documento multimídia.

A interface gráfica da visão espacial do STEVE é apresentada na Figura 5.43. No exemplo desta figura, as características de apresentação do item de mídia que estão sendo exibidas são da mídia *Lion*, visto que este item é a mídia atualmente selecionada (item destacado em verde) na visão temporal. Além dessa integração entre as visões espacial e temporal, na tela de prévia da aplicação hipermídia, são exibidas as mídias cuja duração contém o instante de tempo, no qual a linha indicativa de reprodução está posicionada. Dessa forma, os itens de mídia *Lion* e *Lake* são apresentados na tela de prévia conforme a Figura 5.43.

Abaixo da tela de prévia, existe uma barra de botões que permite ao autor reproduzir

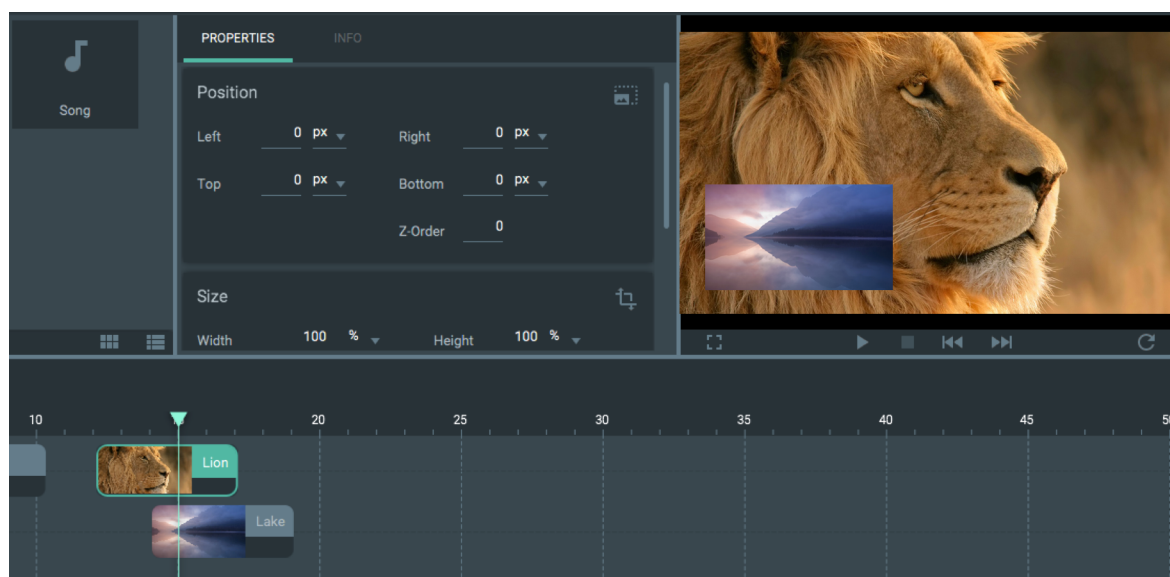


Figura 5.43: Interface gráfica da visão espacial do STEVE

e parar a aplicação multimídia, avançar e retroceder nas cenas, exibir a apresentação multimídia em tela cheia e atualizar a tela de prévia com modificações realizadas nos painéis de propriedades. Apenas os botões de reprodução, de parar e de atualização estão disponíveis nesta versão, 1.0-0. Neste contexto, uma cena é definida como um conjunto de itens de mídias que são apresentados no mesmo intervalo de tempo e, possivelmente, em diferentes regiões da tela. Sempre que um item for adicionado ou removido, uma outra cena é definida.

Baseando-se nas características de apresentação discutidas na Seção 4.5, STEVE oferece diversos painéis de edição de propriedades de acordo com o tipo de arquivo do item de mídia selecionado na visão temporal. Estes painéis são apresentados na aba *PROPERTIES*, sublinhada em verde como mostrado na Figura 5.43. A outra aba, *INFO*, exibe informações básicas da mídia como nome, tipo, instantes de início e fim, duração e se o item é interativo ou não. Esta última aba pode ser visualizada na Figura 5.44. Quando a mídia selecionada é um item do repositório de mídia, os painéis de propriedades não são apresentados e a aba *INFO* exibe somente os campos referentes ao nome e tipo da mídia.

Para os tipos de arquivo imagem, vídeo e texto, STEVE exibe os seguintes painéis de propriedades: *Position*, *Size* e *Style*, os quais representam as entidades *PositionProperty*, *SizeProperty* e *StyleProperty* respectivamente, conforme discutido na Seção 4.5. Desta forma, estes painéis apresentam campos que correspondem aos atributos destas entidades. Os campos são exibidos com valores padrões, ou seja, para itens de mídia que ainda não sofreram nenhuma edição de suas características de apresentação, o sistema define valores iniciais para os campos. Em relação ao painel de posição, todos os campos possuem o

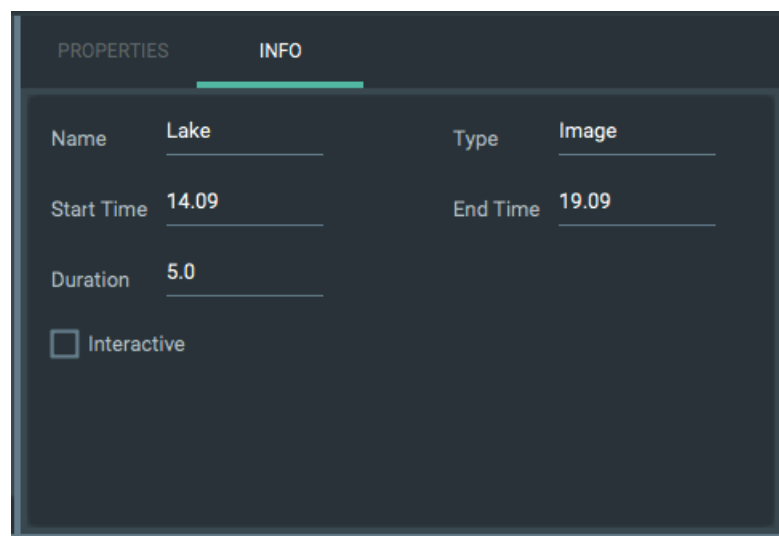
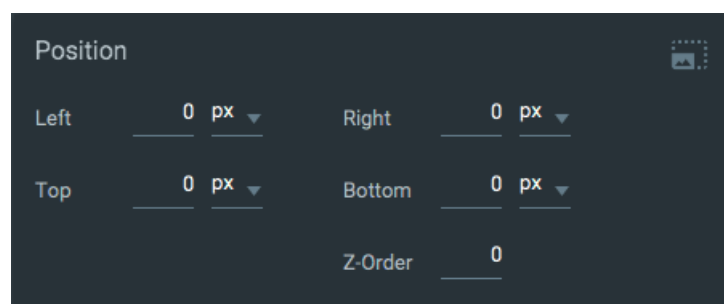
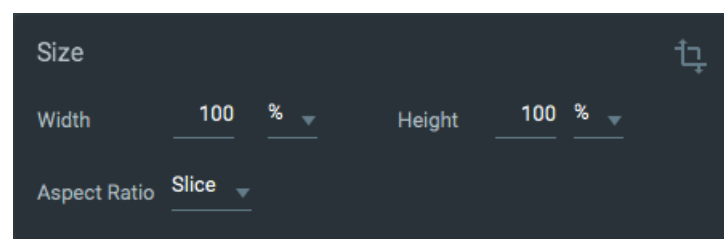


Figura 5.44: Aba INFO de um item de mídia

valor zero. Para o painel de tamanho, a largura e altura da mídia se inicia em 100% e o campo de proporção de tela é configurado, a princípio, com o valor *Slice*. Já no painel de estilo, o campo de transparência possui o valor 0% inicialmente. Todos estes painéis podem ser visualizados na Figura 5.45.



(a) Posição



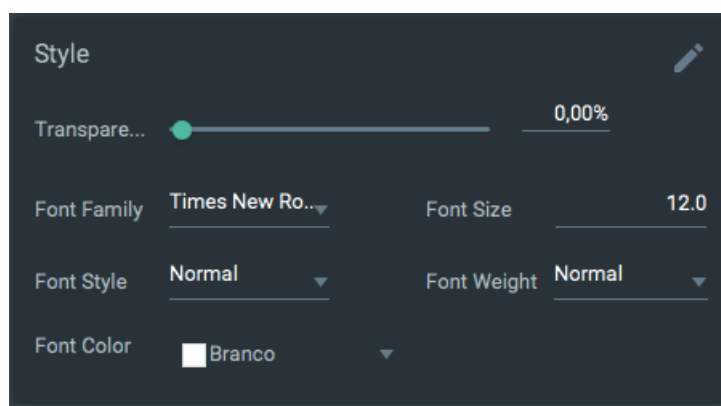
(b) Tamanho



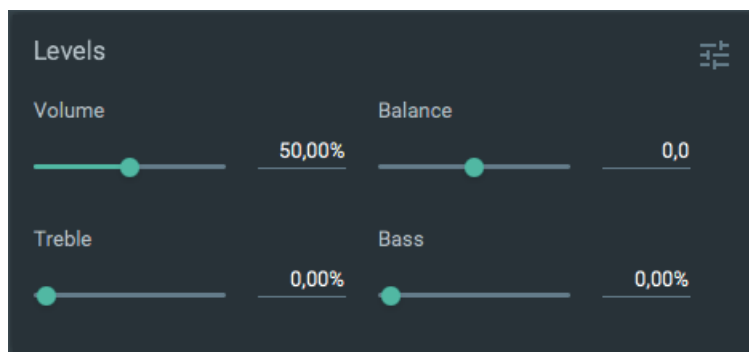
(c) Estilo

Figura 5.45: Painéis de propriedades dos tipos imagem, vídeo e texto

O painel de estilo apresenta propriedades adicionais quando a mídia é do tipo texto, de acordo com o modelo SIMM. Desta forma, o painel, que é ilustrado na Figura 5.46a, passa a exibir campos que referenciam o tipo, tamanho, estilo, espessura e cor da fonte, além da transparência. Estes também são iniciados com valores padrões que correspondem a *Times New Roman*, 12, *Normal*, *Normal*, *white* e 0% respectivamente. Em relação às mídias do tipo áudio e vídeo, o painel *Levels*, que se baseia na entidade *LevelProperty* como enunciado na Seção 4.5, é apresentado. Assim, ele oferece campos para editar o volume, o equilíbrio do som e o nível de agudo e grave como mostrado na Figura 5.46b. O volume se inicia com o valor 50% e os demais campos em zero.



(a) Posição



(b) Tamanho

Figura 5.46: Painéis de propriedades dos tipos imagem, vídeo e texto

## 5.5 Conclusão

Este capítulo apresentou o editor gráfico da visão espaço-temporal para autoria de documentos hipermídia, STEVE. Discutiu suas características gerais, objetivos, interface gráfica e sua arquitetura, mostrando suas entradas e saídas, as estruturas de dados e pacotes usados em sua implementação. Ademais, o repositório de mídias oferecido pelo editor foi apresentado.

O capítulo também detalhou as funcionalidades da edição da visão temporal e espacial do STEVE. Em relação à visão temporal, estes recursos correspondem à adição e remoção de mídias da cadeia temporal, à alteração da duração dos itens e ao reposicionamento deles ao longo da cadeia temporal. Além disso, a sincronização temporal oferecida pela ferramenta proposta foi estudada em detalhes, fazendo uso de figuras que ilustram cada uma das relações temporais síncronas oferecidas. Também foi estudada, a inteligência implementada pelo editor para tratar cada uma das edições feitas pelo usuário que podem prejudicar a consistência temporal. Isto garante que, no momento da execução em alguma plataforma, as aplicações multimídia interativas se comportem de acordo com a sincronização temporal definida pelo usuário na fase de autoria do documento hipermídia.

O suporte à definição de relações de interatividade na edição dos documentos hipermídia, uma das principais contribuições deste trabalho, também foi apresentada neste capítulo, mostrando como o autor especifica este tipo de relação e como ela é representada graficamente e editada na ferramenta. Foi descrita também, a simulação dos eventos interativos definidos na aplicação hipermídia, a fim de facilitar a visualização do comportamento do documento dadas as ocorrências desses eventos. Tanto a definição das relações síncronas quanto as assíncronas são recursos oferecidos por STEVE para evitar as limitações do paradigma de linha do tempo utilizado, principalmente, pelas ferramentas comerciais e também facilitar o uso do modelo baseado em eventos na sincronização temporal dos documentos hipermídia.

No final do capítulo, a visão espacial foi apresentada, discutindo como ela, integrada à visão temporal, permite definir as características de apresentação dos itens de mídia das cadeias temporais e oferece a prévia das aplicações. Note que a visão espacial do STEVE, em sua versão atual, ainda não implementa a sincronização espacial proposta no Capítulo 4.

O próximo capítulo apresentará a arquitetura do software, mostrando o modelo e padrões de projeto utilizados em sua implementação, além de discutir seu diagrama de pacotes e classes enfatizando suas principais funcionalidades. E ainda analisa como STEVE suporta as linguagens NCL e HTML5.



# Capítulo 6

## Implementação

Este capítulo apresenta, na Seção 6.1, a arquitetura de software do STEVE, discutindo seu modelo de implementação e como seus módulos se comunicam entre si. Em seguida, na Seção 6.2, os pacotes e classes que implementam a interface gráfica da ferramenta proposta são analisados. Em relação às classes e pacotes do modelo de dados, estes foram apresentados durante o estudo do SIMM proposto no Capítulo 4. A Seção 6.3 discute os pacotes e classes que são responsáveis pelo suporte da ferramenta às linguagens NCL e HTML5. Além disso, o processo de exportação e importação de documentos NCL como também a geração de aplicações em HTML5 são detalhados nesta seção.

### 6.1 Arquitetura de Software

A arquitetura de software da ferramenta proposta utiliza o modelo orientado a objetos, MVC (*Model-View-Controller*), e é apresentada através do diagrama de pacotes mostrado na Figura 6.1, ilustrando como a comunicação é realizada entre o *front-end* e *back-end* do STEVE.

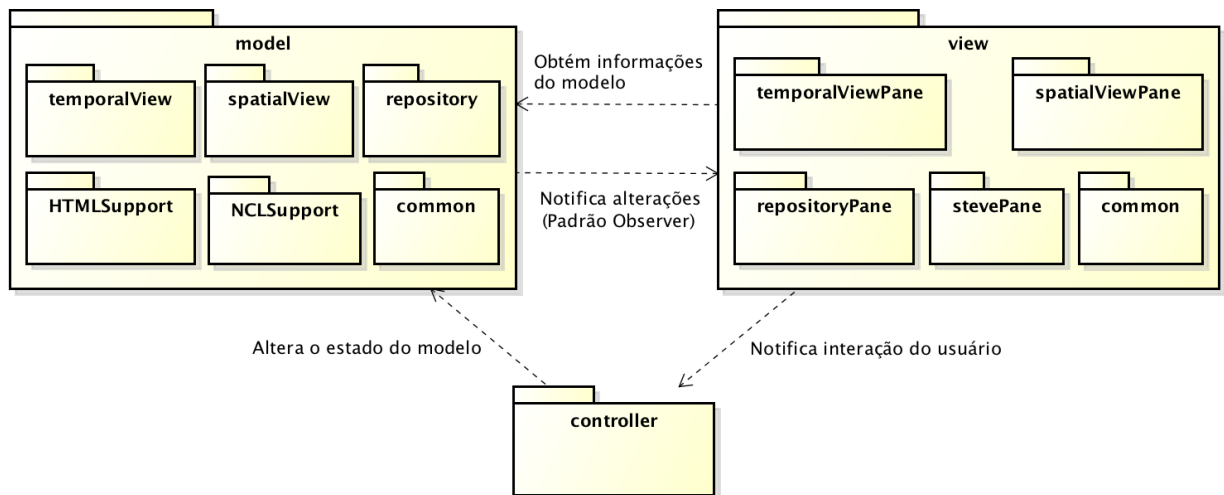


Figura 6.1: Arquitetura do STEVE

O pacote *view* é responsável pela construção da interface gráfica do editor. Ele contém os seguintes pacotes: *temporalViewPane*, *spatialViewPane*, *repositoryPane* e *stevePane*. Os dois primeiros pacotes são responsáveis pela criação da representação gráfica da visão temporal e espacial respectivamente. O pacote *repositoryPane* realiza a implementação da interface do repositório de mídias. E o último pacote é encarregado de construir o leiaute da ferramenta e sua barra de menu.

No pacote *model*, existem os pacotes *temporalView*, *spatialView*, *repository* e *common* que contêm classes que implementam as entidades do modelo SIMM, apresentado no Capítulo 4, além de possuírem toda a regra de negócio da ferramenta. Ou seja, o pacote *temporalView* possui todas classes apresentadas na Figura 4.3, exceto a classe *Media*, que está contida no pacote *common* dentro do *model*. Este pacote armazena classes comuns aos outros pacotes do modelo como as classes *Relation* e *SpatialTemporalView*. Ele ainda contém o pacote *enums* que possui enumerações utilizadas pelas classes do modelo. Cada um dos pacotes do modelo também contém um conjunto de enumerações utilizadas em suas respectivas implementações. Já o pacote *repository* possui apenas uma classe que representa a lista de mídias importadas na ferramenta. E o pacote *spatialView* contém as classes mostradas no diagrama 4.5. Os pacotes *HTMLSupport* e *NCLSupport* serão abordados na Seção 6.3. O diagrama de classes completo do modelo pode ser visualizado no Apêndice A.

O pacote *controller* é responsável por receber notificações de interação do usuário a partir do pacote *view* e alterar o estado do modelo de acordo com estas modificações feitas na visão. Quando o pacote *model* é alterado, este envia notificações de alterações de seu estado para o pacote *view* utilizando o padrão *Observer*. Além disso, sempre que

necessário, os objetos da visão consultam informações sobre o estado do modelo nesta arquitetura.

A ferramenta é implementada utilizando a tecnologia *JavaFX* [15] para oferecer uma interface gráfica rica e ter maior facilidade na manipulação de mídias. Além disso, esta tecnologia possibilita que a ferramenta seja executada em diferentes plataformas.

Nas seções a seguir, os pacotes presentes em *view* serão discutidos detalhadamente. Em seguida, o capítulo irá abordar a implementação do suporte às linguagens NCL e HMTL 5.

## 6.2 Pacotes da Interface Gráfica

Cada um dos pacotes da interface gráfica (pacote *view*) possui outros dois, *images* e *styles*. Estes são responsáveis por armazenar as imagens, utilizadas na interface gráfica dos componentes de tela, e as folhas de estilos, que implementam a formatação da aparência e leiaute da interface. Para simplificar os diagramas seguintes, estes dois pacotes serão omitidos. Dentro do pacote *view*, existe o pacote *common*. Nele, existem ainda os pacotes *fonts* e *languages*. Eles armazenam respectivamente fontes e arquivos de configuração de idiomas utilizados na implementação do STEVE.

Além disso, este pacote contém as classes *Language*, *MessageDialog*, *InputDialog*, *ReturnMessage* e *SliderButton*, e também as interfaces *Observable* e *Observer*. A classe *Language* implementa o suporte de diferentes idiomas, utilizando os arquivos de propriedade do pacote *languages*. As classes *MessageDialog*, *InputDialog* e *ReturnMessage* são encarregadas de criar as janelas exibidas pela ferramenta, visando alertar os usuários, solicitar dados de entrada e apresentar mensagens de retorno respectivamente. A classe *SliderButton* é responsável por implementar os botões do tipo *slider* utilizados pelos controles exibidos na interface gráfica do STEVE. As interfaces citadas implementam o padrão de projeto *Observer* mencionado na Seção 6.1.

No pacote *view*, existe também o pacote *stevePane*, o qual realiza a implementação do leiaute da janela do STEVE. O diagrama deste pacote pode ser visualizado na Figura 6.2. Ele possui as classes: *Main*, *LoadingWindow*, *StevePane*, *SteveMenuBar*, *OpenEventHandler* e *SaveEventHandler*. A classe *Main* é responsável por inicializar a tela de carregamento implementada pela classe *LoadingWindow* da ferramenta e o controlador do sistema (classe *Controller*), o qual instancia as entidades do modelo necessárias e constrói a interface gráfica da ferramenta através da classe *StevePane*. A barra de menu

do STEVE é implementada pela classe *SteveMenuBar*. As classes *OpenEventHandler* e *SaveEventHandler* são encarregadas de concretizar, respectivamente, as seguintes opções de menu: abrir projetos do STEVE existentes e salvar projetos em andamento.

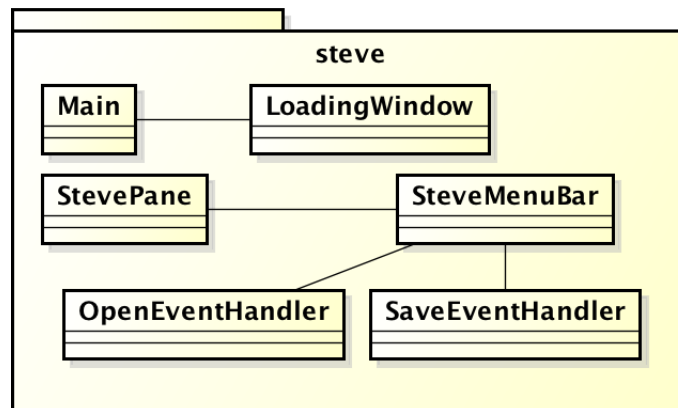
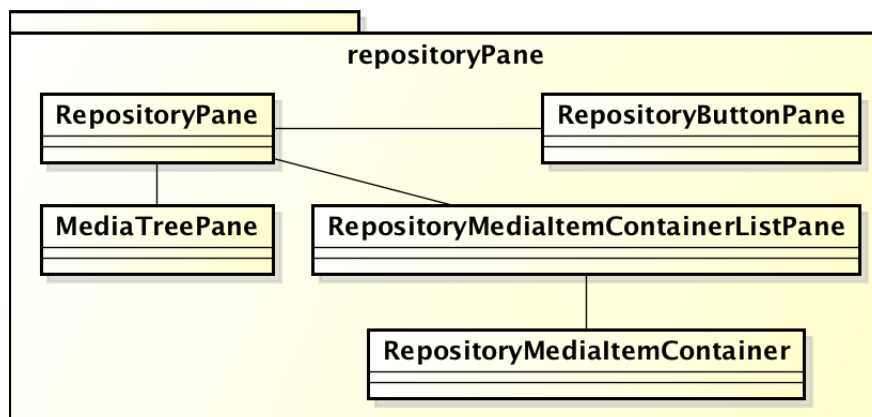


Figura 6.2: Pacote *stevePane* do pacote *view*

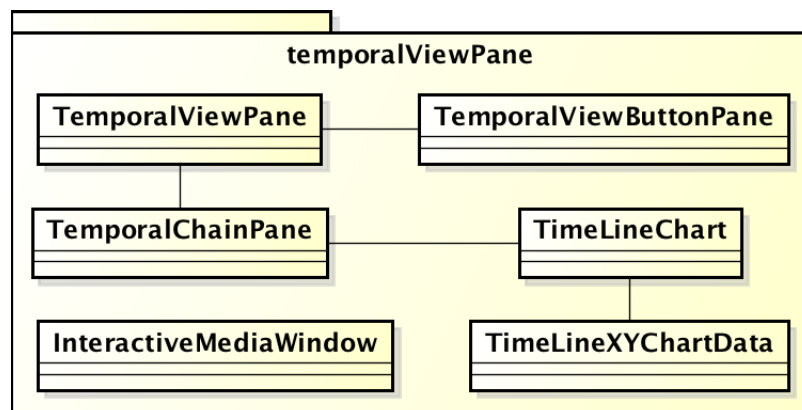
### 6.2.1 Pacote do Repositório de Mídias

O pacote *repositoryPane* mostrado na Figura 6.3 é responsável por criar a interface gráfica do repositório de mídias, apresentado no Capítulo 5. Ele possui as seguintes classes: *RepositoryPane*, *RepositoryButtonPane*, *RepositoryMediaItemContainer*, *RepositoryMediaItemContainerListPane* e *MediaTreePane*. A classe *RepositoryPane* configura a região do repositório de mídias criando os componentes de leiaute necessários para alocar seus componentes, tal como as mídias, a barra de botões e a representação da lista de mídias em uma estrutura de árvore. A classe *RepositoryButtonPane* realiza a implementação da barra de botões. Em relação às classes *RepositoryMediaItemContainer* e *RepositoryMediaItemContainerListPane*, elas são encarregadas de criar o leiaute dos ícones que representam os diferentes tipos de mídia no repositório e de implementar a lista de mídias respectivamente. A classe *MediaTreePane* é responsável por implementar a estrutura em árvore que agrupa os arquivos de mídia por tipo, disponibilizando o formato em lista do repositório de mídia como mostrado na Figura 5.4.

Figura 6.3: Diagrama de classes do pacote *repositoryPane*

### 6.2.2 Pacote da Visão Temporal

No que diz respeito ao pacote da visão temporal, *temporalViewPane*, ele contém classes que implementam a interface gráfica da visão temporal do STEVE. O diagrama de classes deste pacote é mostrado na Figura 6.4. A classe *TemporalViewPane* implementa o leiaute da visão temporal, composto pela barra de botões, pelo gráfico onde os itens de mídia são posicionados no tempo e pelas abas que identificam as cadeias temporais da aplicação hipermídia, conforme ilustrado na Figura 5.6.

Figura 6.4: Diagrama de classes do pacote *temporalViewPane*

A barra de botões é implementada pela classe *TemporalViewButtonPane*, onde as ações de cada botão são definidas. Já o gráfico da visão temporal é implementado pelas classes *TemporalChainPane*, *TimeLineChart* e *TimeLineXYChartData*. A classe *TemporalChainPane* define características iniciais do gráfico como o intervalo do eixo dos tempos e o número de linhas presentes no eixo vertical, além de especificar os eventos interativos que podem ocorrer no gráfico, como o *drag-and-drop*, e de implementar a linha indicativa

da posição do mouse e de reprodução. A classe *TimeLineChart* implementa o gráfico propriamente dito, estendendo a classe *StackedBarChart* do pacote *javafx.scene.chart* [16], o qual oferece facilidades na construção de gráficos em Java. A classe *TimeLineXYChartData* é responsável pela implementação da interface gráfica dos itens de mídia presentes na cadeia temporal. Sobre a classe *InteractiveMediaWindow*, ela implementa a janela de definição das relações de interatividade na ferramenta.

### 6.2.3 Pacote da Visão Espacial

Neste pacote, as classes presentes são encarregadas de implementar a interface gráfica da visão espacial do sistema. O diagrama apresentado na Figura 6.5 contém as classes *SpatialViewPane*, *DisplayPane* e *ControlButtonPane* que são responsáveis, nesta ordem, por definir o leiaute da visão espacial, implementar a tela de prévia e a barra de botões que controlam a reprodução da prévia da aplicação hipermídia. Além disso, a classe *SpatialViewPane* responde a eventos de seleção de mídias tanto no repositório quanto nas cadeias temporais, a fim de exibir os painéis de propriedades adequados de acordo com o tipo da mídia atualmente selecionada na ferramenta.

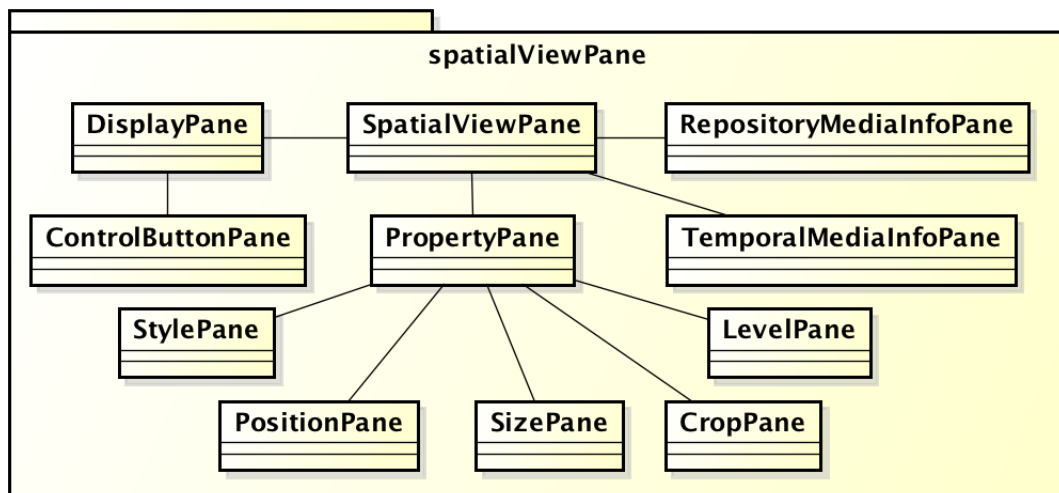


Figura 6.5: Diagrama de classes do pacote *spatialViewPane*

Os painéis de propriedades foram discutidos e ilustrados em detalhes no Capítulo 5. A implementação de cada um dos painéis (*Position*, *Size*, *Crop*, *Style* e *Levels*) é realizada pelas classes *PositionPane*, *SizePane*, *CropPane*, *StylePane* e *LevelPane* respectivamente. A classe *PropertyPane* estabelece o leiaute do conjunto de painéis exibidos na aba de propriedades da visão espacial. Já o painel de informações gerais sobre a mídia selecionada apresentado pela aba INFO é implementado pelas classes *RepositoryMediaIn-*

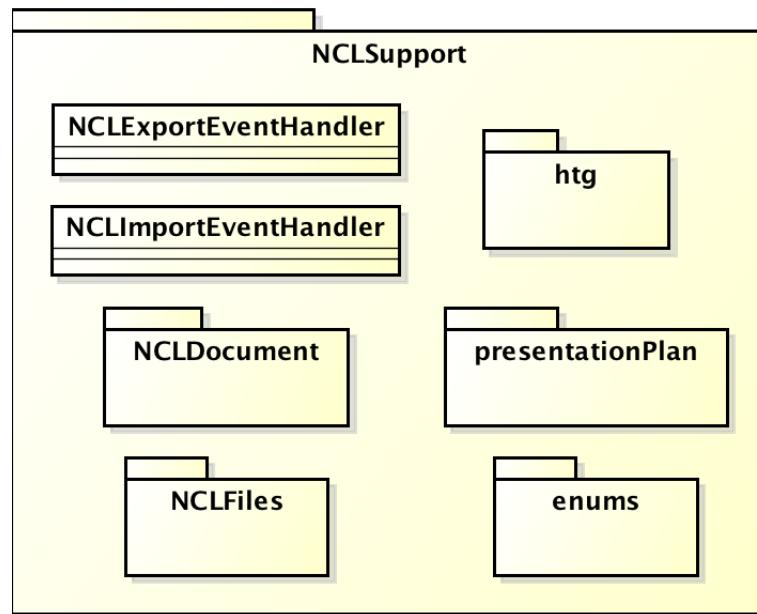
*foPane* e *TemporalMediaInfoPane*, as quais são responsáveis por apresentar os campos adequados quando o item selecionado pertence ao repositório ou a uma cadeia temporal respectivamente.

## 6.3 Suporte à NCL e HTML5

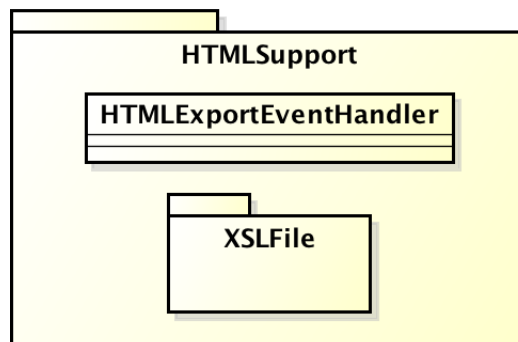
Os pacotes *HTMLSupport* e *NCLSupport*, exibidos no diagrama da Figura 6.6, são encarregados de exportar as aplicações hipermídia para documentos HTML 5 e NCL respectivamente. Em relação ao segundo pacote, ele também é responsável por fornecer as estruturas de dados necessárias para a importação de documentos NCL para o STEVE. Os principais conceitos da linguagem NCL podem ser visualizados no Apêndice B.

A Figura 6.6a apresenta as classes e pacotes que implementam o suporte à linguagem NCL. A classe *NCLExportEventHandler* é responsável por definir a lógica de mapeamento dos objetos Java, entidades do modelo SIMM, que especificam a aplicação hipermídia no STEVE para os elementos da linguagem NCL. Este processo será detalhado na Subseção 6.3.2. Já a classe *NCLImportEventHandler* implementa a importação de documentos NCL para a ferramenta, funcionalidade discutida na Subseção 6.3.1. Em relação aos pacotes que o diagrama 6.6a apresenta, o *HTG* e *presentationPlan* representam as estruturas de dados utilizadas no processo de leitura dos documentos NCL. O pacote *NCLDocument* apresenta classes que estendem entidades da API aNa, a fim de adicionar funções necessárias para implementação da ferramenta proposta. A enumeração dos tipos de conectores presentes na base C, armazenada no pacote *NCLFiles*, são descritas no pacote *enums*.

No que diz respeito ao diagrama 6.6b, que implementa o suporte à linguagem HTML5, contém apenas a classe *HTMLExportEventHandler* e o pacote *XSLFile*. A primeira especifica o processo de transformação da aplicação hipermídia criada no STEVE para um documento HTML5. A lógica desta funcionalidade é apresentada com mais detalhes na Subseção 6.3.2. Esta classe faz uso da API DOM [3] para manipulação de documentos XML e da API *javax.xml.transform* [2] para executar transformações a partir de um documento XML para outro de acordo com a folha de estilo XSL utilizada no processo. Sobre o pacote *XSLFile*, ele possui apenas o arquivo que representa a folha de estilo XSL, denominada *ncl4web.xsl* [1], responsável pela transformação de documentos NCL para o aplicações HTML5.



(a) Suporte NCL



(b) Suporte HTML5

Figura 6.6: Diagramas de pacote para suporte à NCL e HTML5

### 6.3.1 Leitura de Documentos NCL

Nesta seção, será estudado o processo utilizado para importar um documento especificado na linguagem NCL para dentro da ferramenta STEVE. As relações temporais causais são perdidas neste processo, desta forma as mídias são posicionadas diretamente no eixo dos tempos utilizando seus instantes de início e término obtidos através das cadeias temporais geradas a partir das estruturas de dados que serão analisadas a seguir. Para realizar a leitura do documento NCL, STEVE utiliza a API *aNa* [60].

A primeira estrutura corresponde ao HTG [39], *Hypermedia Temporal Graph*, o qual é um modelo de grafo para cadeias temporais hipermédia. A segunda é o plano de apresentação que coleta informações a partir da leitura do HTG necessárias para construir a visão temporal do documento NCL. E a última corresponde à própria cadeia temporal que é derivada do plano de apresentação. É importante enfatizar que as duas últimas



estruturas são deduzidas do *HTG*.

O *Hypermedia Temporal Graph* [63] é um grafo dirigido (dígrafo) que armazena informações necessárias para o controle do sincronismo temporal entre os componentes de um documento NCL. Este dígrafo tem como base a máquina de estado de eventos apresentada na Seção 2.4.1. Ele é definido por um conjunto de vértices e arestas com condições.

Cada vértice é representado pela tripla: nome da ação que dispara a transição na máquina de estado, o identificador da âncora especificando o objeto ou um segmento de seu conteúdo, no caso de eventos de apresentação\seleção ou o identificador da propriedade de uma mídia e seu valor, para eventos de atribuição e o tipo de evento no qual a ação ocorre. Ou seja, o vértice corresponde a transição disparada na máquina de estado para uma determinada âncora sobre um evento.

As arestas do grafo representam os relacionamentos entre as transições, indicando que a transição no vértice de destino somente é disparada se a partir da origem a condição for satisfeita. Estas condições podem ser simples ou compostas. Uma condição simples pode corresponder a um intervalo de tempo que deve ser respeitado para que a transição destino da aresta seja disparada, a uma variável que é testada com um determinado valor que pode ser um intervalo de tempo inclusive e a uma interatividade que deve ocorrer para disparar o vértice destino (transição). Já as condições compostas são duas ou mais condições simples relacionadas com os operadores lógicos *OR*, *AND* e *NOT*.

O processo de transformação do documento NCL para o HTG e a geração do plano de apresentação foram realizados no trabalho [54], no qual tal processo pode ser verificado com mais detalhes. A fim de exemplificar a geração destas estruturas, considere uma aplicação hipermídia que de início exibe um vídeo cuja duração corresponde ao tempo total de apresentação da aplicação. Após alguns segundos de exibição do vídeo, um outro começa ser exibido em tamanho menor sobre o vídeo principal. Quando este último atinge um determinado instante de tempo de exibição, uma imagem que representa um botão é apresentada, indicando interatividade. Se a ação interativa ocorrer, o botão desaparece e é exibida uma nova imagem mostrando informações sobre o vídeo principal. Caso contrário, após um tempo, o botão desaparece, terminando o intervalo de interatividade. A partir disto, somente o vídeo é apresentado até seu término natural. Na Figura 6.7, é mostrada a visão espacial da aplicação.

A Figura 6.8 apresenta o HTG correspondente à aplicação cuja visão espacial é mostrada na Figura 6.7, nela é omitido o tipo de evento *presentation* dos vértices para melhor organização da mesma. O vértice 1 representa o início da exibição do vídeo principal

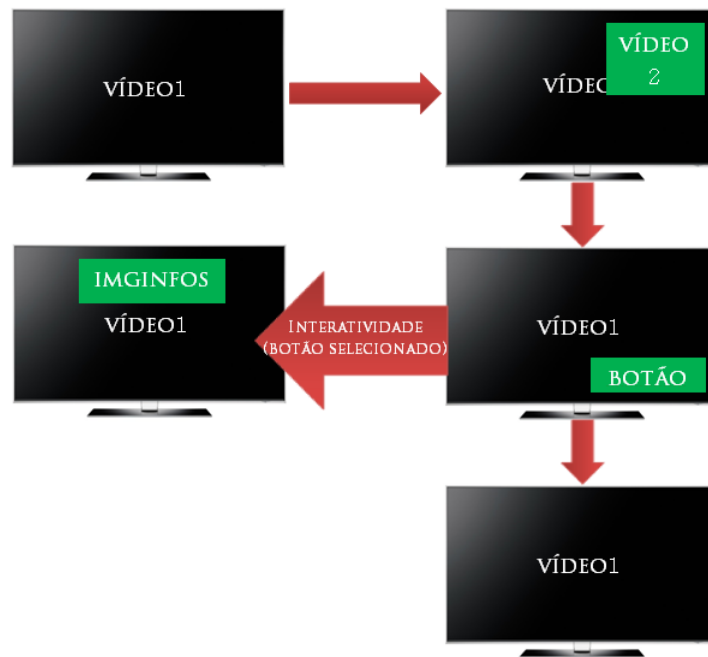


Figura 6.7: Visão espacial da aplicação exemplo

(“vídeo1”). O término do mesmo é indicado pelo vértice 2, sendo que a aresta entre os dois vértices é a duração do vídeo. O vértice 5 representa uma âncora do “video1”, que tem somente o valor de tempo de início (5s), indicando o começo da âncora V (“âncoraV”) que por sua vez determina o início do segundo vídeo (“vídeo2”) representado pelo vértice 6. Já o vértice 7 simboliza o fim do “vídeo2” que possui duração de 10 segundos. Além desta âncora, os vértices 3 e 4 especificam outra com tempo de início (18s) e fim (24s). Ela define o começo e término da mídia “botão” representados pelos vértices 8 e 9 respectivamente. A condição da aresta entre os vértices 8 e 10 indica interatividade que deve ser realizada através do botão vermelho do controle remoto. Após o botão ser selecionado, é iniciada a imagem de informações do vídeo principal (“imgInfos”) que é representada pelo vértice 11 e o término da mídia pelo 12. A mídia “botão”, além de ser finalizada pelo término da “âncoraB”, também é parada pelo fim da apresentação da imagem “imgInfos”. É importante salientar que as arestas cuja condição possui o valor zero indicam que a transição destino deve ser disparada imediatamente quando o vértice origem da condição é atingido.

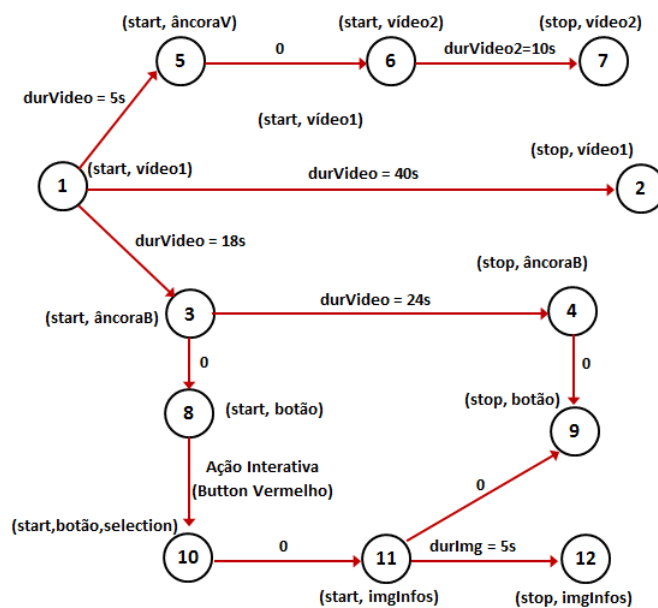


Figura 6.8: HTG para aplicação apresentada na Figura 6.7

A segunda estrutura de dados a ser apresentada, o plano de apresentação [63], é criada a partir do *Hypermedia Temporal Graph* apresentado. Ela é definida como um conjunto de transições de eventos disparados sobre os componentes da aplicação e cada uma destas transições possuem seus instantes de tempo de execução.

Construído o HTG, os valores para os instantes de tempo do disparo de suas transições de evento são calculados a partir do caminho pelas arestas do grafo. Porém, nem sempre esses valores são determinados. Ou seja, para os eventos previsíveis os instantes são completamente determinados antes da execução do documento. Já para os eventos imprevisíveis, isto é, interatividade, os instantes para o disparo das transições de evento de seleção são indeterminados *a priori*. Neste caso, as transições de evento que partem do vértice que representa a interatividade (vértice 10 na Figura 6.8) ou que derivam de vértices que dependem do vértice interativo têm seus tempos calculados relativamente à ocorrência do evento imprevisível.

Por exemplo, na Figura 6.8, os vértices 11, 12 e 9 têm seus instantes de disparo calculados de forma relativa ao momento em que ocorre a ação interativa que dispara o evento de seleção representado pelo vértice 10. Note que neste caso, o vértice 9 é atingido por outro caminho que possui só eventos previsíveis, sendo assim, tem seu tempo determinado *a priori*. A Tabela 6.1 mostra o plano de apresentação para o HTG exibido na Figura 6.8 dividido em duas tabelas: a primeira 6.1a corresponde aos vértices cujas transições têm seu instante de tempo de disparo previsível e a segunda 6.1b apresenta as transições derivadas da ocorrência da interação do espectador.

Transições	Instante de Tempo de Disparo
(start,vídeo1,presentation)	0s
(stop,vídeo1,presentation)	40s
(start,âncoraV,presentation)	5s
(start,vídeo2,presentation)	5s
(stop,vídeo2,presentation)	15s
(start,âncoraB,presentation)	18s
(stop,âncoraB,presentation)	24s
(start,botão,presentation)	18s
(stop,botão,presentation)	24s

(a) Transições Previsíveis

Transições	Instante de Tempo de Disparo
(start,botão,selection)	Xs
(stop,botão,presentation)	(X+0)s
(start,imgInfos,presentation)	(X+0)s
(stop,imgInfos,presentation)	(X+5)s

(b) Transições Imprevisíveis

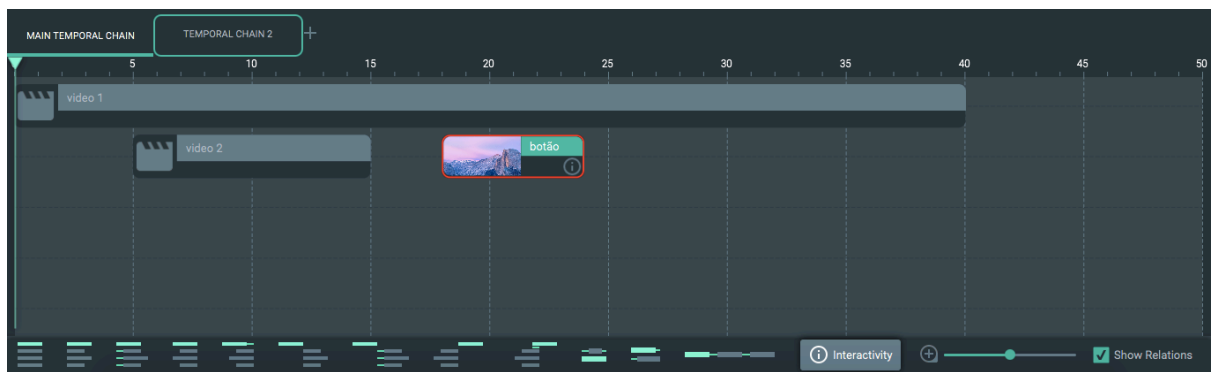
Tabela 6.1: Plano de apresentação para o HTG da Figura 6.8

Desta forma, uma parte do plano de apresentação possui todas as transições de eventos que têm seus instantes de tempo de disparo previsíveis e outra com as transições que possuem tempos calculados relativos ao tempo “X” em que ocorreu a ação interativa que selecionou, no exemplo, a mídia “botao”.

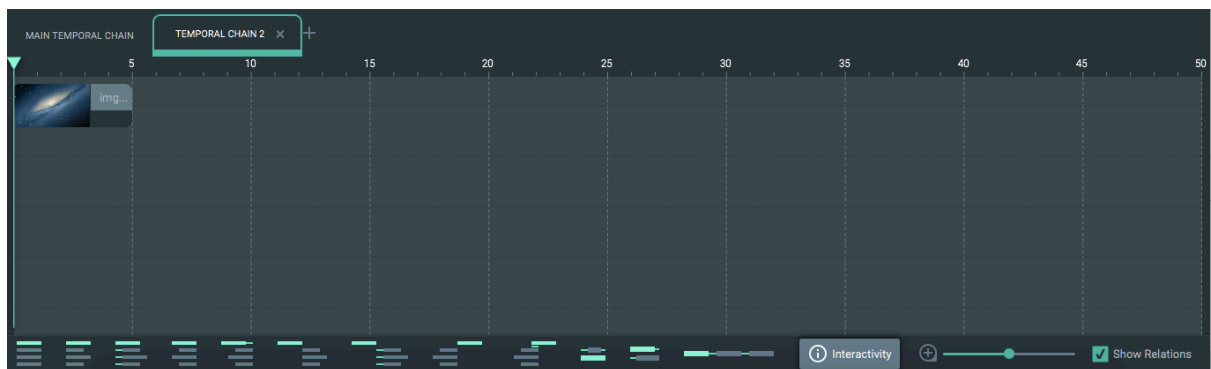
A partir do plano de apresentação especificado na Tabela 6.1, é possível determinar os instantes de tempo em que os componentes da aplicação são exibidos, ou seja, construir as cadeias temporais da aplicação. Uma cadeia temporal armazena, para cada mídia que pode ser apresentada na execução da aplicação, seu instante de início e término de apresentação. Para algumas mídias (discretas), o tempo de fim pode não ser determinado, pois na fase de autoria do documento hipermídia, o fim da mídia pode não ter sido especificado através de uma duração explícita (tempo definido pelo autor) e nem por um elo que a finaliza.

No caso de interatividade, algumas mídias podem ser finalizadas e outras novas apresentadas após a ação interativa ocorrer. Sendo assim, além dos tempos de início e fim das novas mídias serem armazenados na cadeia temporal, o tempo de parada dos objetos que são finalizados também devem ser guardados na cadeia. A Figura 6.9 mostra as cadeias temporais geradas pelo STEVE ao importar o documento NCL exemplo.

A Figura 6.9a mostra a cadeia temporal do documento hipermídia, onde o item de



(a) Cadeia temporal principal



(b) Cadeia temporal secundária derivada da interação

Figura 6.9: Visão temporal para aplicação da Figura 6.7

mídia “vídeo1” inicia a apresentação no instante 0s e a finaliza em 40s. No tempo 5s, o segundo vídeo é iniciado e dura até quinze segundos. No intervalo de tempo 18s a 24s (intervalo de interatividade), a mídia “botão” é apresentada indicando possibilidade de interação. Caso o “botao” não seja selecionado, esta mídia é terminada no instante de tempo 24s e somente o “vídeo1” é apresentado na tela até seu fim natural (40s). Além disso, a relação de interatividade está representada graficamente nesta figura da seguinte maneira. O item “botão” está destacado com borda vermelha indicando que ele será finalizado quando a interação ocorrer e a cadeia *Temporal Chain 2*, com borda verde, informando que ela será iniciada. Desta forma, esta cadeia, ilustrada na Figura 6.9b, exibe apenas a mídia “imgInfos”, a qual é iniciada imediatamente após o evento interativo ocorrer e dura por 5s.

### 6.3.2 Exportando para NCL e HTML 5

Dado que o modelo SIMM, utilizado por STEVE para especificar as aplicações hiper-mídia, é independente de linguagem de autoria multimídia e apresenta uma modelagem simples, principalmente, na definição das relações temporais, a exportação das aplicações












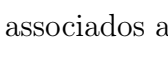
hipermídia criadas na ferramenta para outro modelo é facilitada.

Para realizar a transformação da aplicação hipermídia para um documento NCL, as entidades do modelo SIMM devem ser mapeadas para os elementos da linguagem NCL da seguinte maneira. Primeiramente, a mídia mestre da cadeia temporal principal do projeto STEVE é mapeada através de um elemento NCL `<port>`, de forma que esta inicie o documento NCL. Para cada mídia das cadeias temporais presentes na aplicação desenvolvida na ferramenta proposta, o sistema especifica um elemento `<media>`, ao qual os elementos `<region>` e `<descriptor>` estão associados. Para criar o elemento `<region>`, os valores estabelecidos nos painéis de propriedade *Position* e *Size* são utilizados da seguinte maneira. Os campos *Left*, *Right*, *Top*, *Bottom* e *Z-Order* do painel *Position* são mapeados para os atributos da região *left*, *right*, *top*, *bottom* e *zIndex* respectivamente. E os campos *Width* e *Height* do painel *Size* são utilizados para definir os valores dos atributos *width* e *height* de `<region>` nesta ordem. Já para criação do elemento `<descriptor>`, os campos *Aspect Ratio* do painel *Size* e *Transparency* da área *Style* são utilizados para especificar os valores dos atributos *fit* e *transparency* do descritor NCL respectivamente.

Após a criação das mídias e de suas regiões e descritores, além das portas do documento NCL, o sistema realiza a definição dos elos temporais. Para isso, cada uma das relações oferecidas na visão temporal do STEVE, como discutido na Seção 5.3, deve ser mapeada para elos NCL (elemento `<link>`) cuja semântica da relação é definida por conectores hipermídia [56]. Assim, STEVE utiliza uma base de conectores predefinida para especificar os elos NCL, de acordo com o tipo de relação temporal do modelo SIMM que está sendo mapeada no momento da exportação. A base de conectores completa está disponível no Apêndice C. Além de especificar o atributo `<xconnector>` do elemento `<link>`, os elementos `<bind>` devem ser especificados para definir qual item de mídia vai estar ligado à condição e às ações do elo. E caso algum valor de atraso seja definido para relação temporal, também deve ser definido o elemento `<linkParam>` para representar este atraso. A Tabela 6.2 mostra a correspondência entre as relações temporais do STEVE e os conectores hipermídia desta base que são utilizados para definição dos elos NCL.

Para enfatizar a simplicidade da definição de uma relação temporal no modelo SIMM quando comparada à especificação correspondente em NCL, pode-se analisar a relação *Starts* como exemplo. Para defini-la, deve-se especificar somente uma mídia mestre, a lista de escravas e o tipo da relação, no caso *Starts*, como a Figura 6.10a ilustra. Por outro lado, a especificação em NCM e NCL envolve a definição de um conector hipermídia que dá a semântica da relação temporal, indicando a condição e a ação do elo. Além disso,

Tabela 6.2: Relações Temporais Oferecidas pelo STEVE

Relação Temporal	Símbolo	Conector Hipermídia
<i>Starts</i>		<i>onBeginStart</i>
<i>Starts With Delay</i>		<i>onBeginStartDelay</i>
<i>Finishes</i>		<i>onEndStop</i>
<i>Finishes With Delay</i>		<i>onEndStopDelay</i>
<i>Meets</i>		<i>onEndStart</i>
<i>Meets With Delay</i>		<i>onEndStartDelay</i>
<i>Met By</i>		<i>onBeginStop</i>
<i>Met By With Delay</i>		<i>onBeginStopDelay</i>
<i>During</i>		<i>onBeginStopDelay + onBeginStartDelay</i>
<i>Overlaps</i>		<i>onBeginStartDelay + onEndStopDelay</i>
<i>Equals</i>		<i>onBeginStart + onEndStop</i>
<i>Before</i>		<i>onEndStartDelay</i>

os itens de mídia devem ser associados aos papéis de condição e ação do conector do *link* através de elementos *<bind>*. A Figura 6.10b mostra os componentes necessários para especificar a relação *Starts* na linguagem NCL.

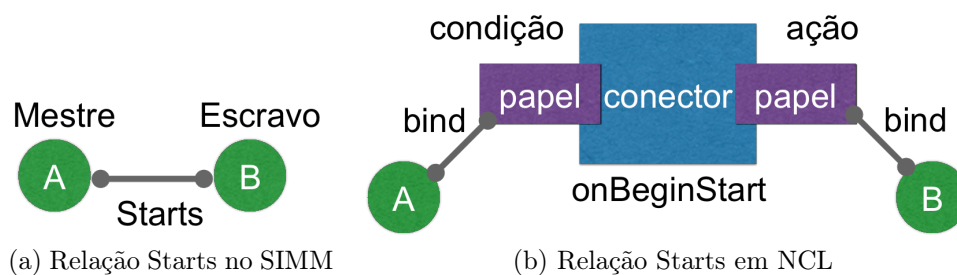


Figura 6.10: Especificação da relação Starts no modelo SIMM e em NCL

É importante observar que a relação *Before*, apresentada na última linha da tabela, necessita especificar diversos elos que utilizam o conector *onEndStartDelay*. A fim de ilustrar este funcionamento, a Figura 6.11 exibe os elos criados destacando seus respectivos itens participantes. Para a primeira mídia escrava da relação (segunda mídia da sequência de itens que formam a relação *Before*), a condição do elo está associada ao término da mídia mestre da relação *Before*, isto é, primeira mídia da sequência de itens. Satisfeita

esta condição, a ação do elo é disparada, a qual inicia, neste caso, a primeira mídia escrava após o atraso estabelecido pela relação. Para a segunda mídia escrava da relação (terceira mídia da sequência), é criado um elo utilizando o mesmo conector, entretanto, com a condição associada ao término da primeira mídia escrava e a ação se referindo ao início desta segunda escrava. E assim é feito, sucessivamente, para as demais mídias da sequência.

```
<link id='link_10' xconnector='connectorBase#onEndStartDelay'>
  <linkParam name='delay' value='1.0s' />
  <bind role='onEnd' component='Lion' />
  <bind role='start' component='Yosemite' />
</link>
<link id='link_11' xconnector='connectorBase#onEndStartDelay'>
  <linkParam name='delay' value='1.0s' />
  <bind role='onEnd' component='Yosemite' />
  <bind role='start' component='Lake' />
</link>
<link id='link_12' xconnector='connectorBase#onEndStartDelay'>
  <linkParam name='delay' value='1.0s' />
  <bind role='onEnd' component='Lake' />
  <bind role='start' component='Beach' />
</link>
```

Figura 6.11: Elos criados para definir a relação Before em NCL

As relações de interatividade também devem ser mapeadas para elementos NCL na exportação. Dessa forma, para cada relação de interatividade, a ferramenta cria um elo NCL cuja semântica é dada pelo conector *onSelectionStartStop* ou *onSelectionStartStop-Delay*. O primeiro conector é utilizado quando nenhum atraso é especificado na definição da relação para disparar suas ações. Caso contrário, o segundo conector estabelece a causalidade do elo de interatividade. Em ambos os conectores, a condição corresponde à seleção de um item (mídia interativa) pelo espectador, para então, iniciar novas cadeias temporais ou terminar itens de mídia que estejam sendo apresentados durante a interação do usuário.

Além da especificação desses elementos descritos acima, o sistema verifica se algum item de mídia do documento não possui seu instantes de início definido. Esta especificação pode ser dada por um elo que inicie a mídia, por exemplo a relação *Meets*, ou definida pelo item ser o mestre de uma cadeia temporal, visto que neste caso, a mídia é mapeada através de uma porta do documento NCL. Caso o início de uma mídia não esteja determinado, a ferramenta cria um elo NCL para iniciar esta mídia. Este faz uso do conector *onBeginStartDelay* cuja condição está associada ao instante de início da mídia mestre da cadeia temporal, à qual o item sem definição de início pertence. E a ação deste conector corresponde ao início deste item. Além disso, o sistema obtém o atraso entre o início da mídia mestre da cadeia temporal e o início do item que foi apenas posicionado



diretamente na linha do tempo pelo usuário sem estabelecer nenhuma relação de definição do início de sua apresentação.

Outra validação que é realizada por STEVE, durante a exportação, refere-se a identificar se a mídia mestre da cadeia temporal principal não está iniciando no instante zero. Caso isso se confirme, o sistema apresenta duas opções ao usuário: manter o instante de início da mídia ou reposicioná-la no instante 0s. Na primeira opção, o sistema criará um temporizador no documento NCL (elemento `<media>` sem conteúdo) e o associa ao elemento `<port>` para iniciar o documento. Além disso, cria-se uma âncora temporal no temporizador que corresponde ao atraso utilizado para iniciar o nó de mídia, o qual representa o mestre da cadeia principal. Entretanto, quando o usuário seleciona a segunda opção, o sistema automaticamente reposiciona a mídia para o instante 0s, associando-a à porta do documento NCL.

As mídias contínuas que tiveram suas respectivas durações naturais alteradas pelo usuário durante a autoria no STEVE, seja através de um elo ou da definição explícita da duração, possuem o seguinte tratamento durante a exportação NCL. Para mídias do tipo áudio, quando a duração implícita é menor que o intervalo de apresentação definido pelo autor, o sistema cria um elo NCL que faz uso do conector *onEndStart* cuja condição é o término desta mídia e a ação corresponde ao seu próprio início, assim, colocando a mídia em *loop*. Neste caso, se não existe nenhum elo temporal que a finalize, o sistema cria um temporizador que possui uma âncora temporal cujo término corresponde ao fim da mídia em *loop* e inicia junto com a aplicação. Além disso, é criado um elo cuja semântica é dada pelo conector *onEndAbort*. Seu papel de condição se refere ao término da âncora do temporizador e sua ação é definida pelo aborto do item em *loop*. Entretanto, quando existe um elo na aplicação criada no STEVE que finaliza esta mídia, ao invés de fazer uso do temporizador, o sistema utiliza a ação *abort* neste elo, modificando o mapeamento definido na Tabela 6.2. Note que caso fosse usada a ação *end*, a mídia continuaria em laço. Ainda no caso onde a duração implícita do áudio é menor que a duração definida pelo autor, todas as condições de elo que correspondem ao término da mídia em *loop* passam a estar associadas com o término da âncora do temporizador, que de fato representa o fim da mídia.

Para ilustrar este processo na exportação, considere uma aplicação multimídia onde existe apenas uma mídia do tipo áudio que possui 15s de duração. Esta mídia, entretanto, possui apenas 10s de duração implícita. Assim, durante a exportação para NCL, foi criado um temporizador, mídia *temp*, com uma âncora finalizando em 15s conforme o código 6.1.

Ademais, dois elos foram especificados. O primeiro, *link\_1*, estabelece o *loop* da mídia áudio e o segundo, *link\_2*, define o disparo da ação *abort* sobre o áudio quando a âncora *area1* termina no instante 15s.

Listing 6.1: Documento NCL exportado para o exemplo de repetição da mídia áudio

```

1 <?xml version='1.0' encoding='ISO-8859-1'?>
2 <ncl>
3   <head>
4     <regionBase>
5       <region id='rg_audio' left='0' right='0' top='0' bottom='0' height='100.0%'
6         width='100.0%' />
7     </regionBase>
8     <descriptorBase>
9       <descriptor id='desc_audio' region='rg_audio' />
10    </descriptorBase>
11    <connectorBase>
12      <importBase alias='connectorBase' documentURI='causalConnectorBase.ncl'
13        baseId='causalConnectorBase' />
14    </connectorBase>
15  </head>
16  <body>
17    <port id='port_audio' component='audio' />
18    <port id='port_temp' component='temp' />
19    <media id='audio' src='media/Song.mp3' descriptor='desc_audio' />
20    <media id='temp'>
21      <area id='area1' end='15s' />
22    </media>
23    <link id='link_1' xconnector='connectorBase#onEndStart'>
24      <bind role='onEnd' component='audio' />
25      <bind role='start' component='audio' />
26    </link>
27    <link id='link_2' xconnector='connectorBase#onEndAbort'>
28      <bind role='onEnd' component='temp' interface='area1' />
29      <bind role='abort' component='audio' />
30    </link>
31  </body>
32 </ncl>

```

Quando a mídia contínua é um vídeo e sua duração implícita é menor que o intervalo de apresentação definido pelo autor, STEVE exibe uma janela apresentando duas opções para o usuário. A primeira refere-se em repetir o conteúdo do vídeo até atingir a duração da mídia. Para isso, a exportação para NCL é feita de forma análoga a mídias do tipo

áudio. A outra opção corresponde em congelar o último quadro do vídeo até alcançar a duração da mídia. Neste caso, a ferramenta define o valor *true* ao atributo *freeze* do elemento *descriptor* que está associado ao vídeo. Caso não exista um elo na aplicação que não o finalize, o sistema especifica a duração definida pelo autor no atributo *explicitDur* deste mesmo elemento.

Em ambos os tipos de mídia, áudio e vídeo, quando a duração implícita é maior que a duração definida pelo autor, o item de mídia é apresentado até o fim desta última, interrompendo a reprodução do conteúdo da mídia. Para isso, o sistema apenas especifica a duração da mídia no atributo *explicitDur* do elemento *descriptor* durante a escrita do documento NCL, caso não exista qualquer elo temporal criado na fase de autoria que finalize esta mídia. Diante desses tratamentos realizados pelo sistema durante a exportação, o documento NCL gerado apresenta o mesmo comportamento descrito na Subseção 5.3.1 em relação à alteração da duração de mídias contínuas.

Ao final do mapeamento da aplicação hipermídia criada na ferramenta proposta para elementos da linguagem NCL, o documento gerado é salvo em um diretório selecionado pelo usuário, onde também uma pasta, nomeada de *media*, é criada contendo todos os arquivos de mídia utilizados pela aplicação. Para a criação do documento NCL, é utilizada a API *aNa* [60].

Além do STEVE realizar a exportação de suas aplicações para documentos NCL, ele também gera documentos em HTML5. Para isso, o sistema utiliza a ferramenta NCL4WEB [61], que permite executar as aplicações hipermídia NCL na Web. A implementação desta ferramenta faz uso das tecnologias XSLT e JavaScript para transformar o código especificado na linguagem NCL em um documento HTML5. Assim, primeiramente, STEVE gera o documento NCL da aplicação hipermídia e, em seguida, o salva no diretório *home* do usuário temporariamente. Depois, o sistema recupera este documento NCL e o transforma em um documento HTML5, aplicando a folha de estilo oferecida por NCL4WEB [61]. O documento HTML5 é salvo no *home* do usuário juntamente com a pasta que contém todos os arquivos de mídia referenciados pela aplicação.

Devido a limitações do NCL4WEB, o documento NCL intermediário é gerado sem utilizar uma base de conectores externa. Isto é, todos os conectores são definidos no próprio documento NCL. Além disso, o atributo *freeze* do elemento *<descriptor>* não é interpretado pelo NCL4WEB. Consequentemente, a especificação do autor para congelar o vídeo, quando sua duração implícita é menor que a duração definida na aplicação, não é refletida no documento HTML5. É importante destacar que todas as relações tem-

porais, síncronas e assíncronas, oferecidas no STEVE são suportadas pelo NCL4WEB. Entretanto, nem todas as características de apresentação disponíveis na ferramenta são suportadas na transformação do NCL4WEB. As propriedades que correspondem aos painéis de posição, tamanho e estilo, vistos nas Figuras 5.45a, 5.45b e 5.45c respectivamente, são traduzidas para o HTML5. Porém, as propriedades particulares às mídias do tipo áudio e texto não são suportadas pela folha de estilo *ncl4web.xsl*.

## 6.4 Conclusão

Este capítulo apresentou a arquitetura de software do STEVE, discutindo seu modelo orientado a objetos, o MVC (*Model-View-Controller*), e como ocorre a comunicação entre os módulos da arquitetura. Esta forma de implementação, onde o modelo é desacoplado da camada de visualização (interface gráfica), facilita a manutenção e atualização da ferramenta, visando estender suas funcionalidades. O diagrama de pacotes e classes do STEVE também foi discutido em detalhes, apresentando, separadamente, os pacotes que implementam a interface gráfica da ferramenta.

Em seguida, o capítulo focou no suporte à NCL e HTML5 dado pela ferramenta. Além de apresentar o diagrama de pacotes e as classes que implementam este suporte, o capítulo descreveu como o sistema realiza a leitura de documentos NCL visando a importação para o STEVE. Para isso, ele apresentou como são definidas as estruturas de dados utilizadas na leitura de documentos NCL, ilustrando com exemplos. Finalizando o capítulo, a exportação para NCL e HTML5 foi estudada. O processo realizado pelo sistema para mapear as entidades do modelo SIMM, o qual representa as aplicações hipermídia no STEVE, para elementos da linguagem NCL foi explicado em detalhes. Ademais, a exportação para HTML5 através da geração de um documento NCL intermediário também foi introduzido no capítulo. As limitações neste processo de transformação para HTML5 também foram levantadas.

O próximo capítulo discute a avaliação da ferramenta STEVE. Para isso, é realizada uma análise comparativa entre os trabalhos relacionados e são apresentados os resultados dos testes de usabilidade e da experiência do usuário aplicados na ferramenta proposta. O capítulo também discute o método utilizado e o teste realizado propriamente dito.

# Capítulo 7

## Avaliação da Ferramenta

Este capítulo apresenta uma análise dos trabalhos relacionados na Seção 7.1. Em seguida, apresenta o teste de usabilidade realizado para avaliar o STEVE na Seção 7.2. A Subseção 7.2.1 discute a metodologia utilizada para realizar o teste de usabilidade. Os questionários que foram utilizados para coletar os *feedbacks* dos usuários são apresentados na Subseção 7.2.2. Já os resultados do teste são analisados na Subseção 7.2.3.

### 7.1 Comparação com os Trabalhos Relacionados

A ferramenta *Composer* não possui, em sua versão final, a edição da visão temporal. Além disso, ela não oferece a visão espacial integrada com a temporal, de forma a permitir a prévia da execução das aplicações hipermídia. A simulação ou execução das aplicações também não pode ser realizada dentro do ambiente do editor. Visto que a ferramenta oferece visões gráficas das aplicações hipermídia baseando-se diretamente nas entidades do modelo NCM, o usuário necessita conhecer a estrutura deste modelo para estar apto a utilizar a ferramenta.

O trabalho *NEXT* permite que autores sem conhecimento de modelo ou linguagem de autoria multimídia consigam criar suas aplicações hipermídia fazendo o uso de templates de composição. Entretanto, para os autores utilizarem outros recursos da ferramenta, como sua visão estrutural, eles devem conhecer o modelo NCM, como ocorre com a ferramenta *Composer*. *NEXT* também não oferece a simulação das aplicações hipermídia, nem a prévia da execução. A visão temporal também não é oferecida no editor.

Tanto a ferramenta *CMIFed* quanto *GRI*ns utilizam um modelo de sincronização baseado em composições hierárquicas (paralelas e sequenciais) que especifica como os objetos

de mídia dos documentos estão relacionadas no tempo. A visão temporal disponibilizada por estes editores apresentam estas composições diretamente na linha do tempo, o que pode dificultar a compreensão da sincronização temporal em aplicações mais complexas, principalmente, por usuários que não possuem conhecimento sobre o modelo de sincronização temporal hierárquico.

Já o sistema *FireFly* apresenta uma visão em grafo bastante expressiva, mas pouco intuitiva para o usuário, principalmente, na verificação da ordem dos itens de mídia da aplicação, visto que os nós não estão dispostos diretamente no tempo. Além de não oferecer a edição da visão espacial.

O sistema *LimSee 2* apresenta diversas visões da aplicação multimídia, inclusive a visão temporal. Entretanto, ele não suporta a definição de relações de interatividade e exige um determinado nível de conhecimento do modelo hierárquico para sincronizar temporalmente os itens de mídia da aplicação.

Além destas ferramentas, foram analisados softwares comerciais que permitem a criação de apresentações multimídia. Eles usam o paradigma de sincronização baseada em *timeline* que é bastante intuitivo e fácil na compreensão da ordem temporal das aplicações. Estas ferramentas também possuem interfaces intuitivas e bem aceitas no mercado, oferecendo a visão temporal integrada à visão espacial. Entretanto, esses editores apresentam sérias limitações que são intrínsecas ao paradigma adotado. Como não suportar a definição de eventos assíncronos (interatividade e testes de variáveis), visto que os instantes de início e término dos componentes da aplicação devem ser conhecidos a priori. Essas ferramentas não permitem definir relações temporais entre os itens de mídia, o que pode comprometer o sincronismo das aplicações. Além disso, elas não produzem documentos hipermídia especificados em uma linguagem padrão de autoria multimídia, por exemplo NCL e HTML5, como formato final da aplicação. Elas geram apenas formatos de vídeo, por exemplo MP4 e MOV, para representar as aplicações multimídia.

Apesar de não permitir a execução das aplicações hipermídia dentro do próprio ambiente de autoria, o editor gráfico STEVE, proposto nesta dissertação, oferece a visão temporal e espacial integradas permitindo a prévia da execução das aplicações hipermídia. Além disso, ele permite que os autores exportem as aplicações criadas na ferramenta para documentos NCL e HTML 5. Assim, usuários podem executá-las no middleware GINGA (TV digital) ou em qualquer navegador Web e, conseqüentemente, em computadores, tablets e smartphones. Na visão temporal, STEVE utiliza relações temporais do modelo SIMM para realizar a sincronização temporal da aplicação, de forma a não utilizar

o paradigma *timeline*, evitando suas limitações. Além disso, permite que relações de interatividade sejam definidas pelos autores e que sejam feitas definições de características de exibição das mídias. Devido à simplicidade do modelo SIMM quando comparado a outros modelos mais complexos da literatura, como o NCM, STEVE suporta a criação de aplicações multimídia interativas por usuários sem conhecimento de linguagens ou modelos multimídia, utilizando uma interface simples baseando-se nas interfaces bem aceitas das ferramentas comerciais.

A Tabela 7.1 resume os recursos discutidos no Capítulo 3, comparando as ferramentas de autoria multimídia, também apresentadas no Capítulo 3, com o editor STEVE.

Tabela 7.1: Recursos das Ferramentas de Autoria

Recurso / Editor	Composer	NEXT	CMIFed	FireFly	GRiNs	LimSee 2	STEVE
Paradigma de autoria	E	E	H	R	H	H	E
Formato final da aplicação	NCL	NCL	SMIL	Modelo FireFly	SMIL	SMIL	NCL / HTML5
Edição da Visão temporal			✓		✓	✓	✓
Edição da Visão espacial	✓	✓	✓		✓	✓	✓
Suporte à interatividade	✓	✓	✓	✓	✓		✓
Prévia da execução			✓	✓	✓	✓	✓
Suporte a autores sem conhecimento de modelo ou linguagem		✓					✓
Execução			✓	✓	✓	✓	

E : Baseado em Eventos; R : Baseado em Restrições; H : Hierárquico;

✓ : contemplado pelo editor

## 7.2 Teste de Usabilidade

O teste de usabilidade do STEVE foi realizado por um grupo de 10 pessoas, sendo que 3 desses usuários nunca tinham utilizado um editor de vídeo. O perfil do grupo se caracteriza, predominantemente, por pessoas profissionais e estudantes na área de computação

correspondendo a 8 usuários dos 10 selecionados para o teste. Dentro deste grupo, apenas dois conheciam a linguagem NCL.

Para realizar este teste, foram selecionadas as medidas de usabilidade globais [4]: eficácia, eficiência e satisfação. Além disso, também foram levadas em consideração a expressividade dos componentes da ferramenta e sua interface gráfica. Foram utilizados dois questionários: o primeiro é o SUS (System Usability Scale) [51] para avaliar a eficácia, eficiência e satisfação do STEVE e o segundo é um formulário específico para mensurar a expressividade dos componentes da ferramenta, a organização da interface gráfica e coletar opiniões gerais sobre o uso do STEVE. É importante enfatizar que o teste de usabilidade foi focado nas funcionalidades da visão temporal do STEVE.

### 7.2.1 Metodologia

O objetivo do teste foi avaliar a facilidade de edição utilizando a visão temporal da ferramenta STEVE na autoria de uma aplicação hipermídia. Dessa forma, a aplicação definida para realizar o teste apresenta poucos objetos de mídia e relacionamentos, a fim de remover a dificuldade na elaboração da aplicação e focar a atenção do usuário na utilização dos componentes da ferramenta.

Inicialmente, alguns conceitos básicos foram passados para os usuários como a definição de uma aplicação multimídia interativa, exemplos de aplicação hipermídia e de ferramentas comerciais bastante comuns na criação de apresentações multimídia. Em seguida, STEVE foi apresentado discutindo seus objetivos e motivação. Além disso, sua interface gráfica foi mostrada indicando suas principais funcionalidades como a adição de mídias na visão temporal, a sincronização temporal através das relações temporais oferecidas pela ferramenta e a definição de mídias interativas.

Após a apresentação geral sobre o assunto e da própria ferramenta, a aplicação hipermídia a ser construída pelos usuários utilizando o STEVE foi descrita em 3 etapas. A primeira possui o seguinte comportamento: um vídeo inicia a apresentação e, após 1s do término do vídeo, toca um áudio. A visão temporal desta primeira etapa pode ser visualizada na Figura 7.1.

Depois do término da primeira etapa, os usuários testaram suas aplicações no navegador Web utilizando a funcionalidade do STEVE de exportação da aplicação hipermídia para um documento HTML5. A segunda etapa da aplicação do teste, cuja visão temporal é ilustrada na Figura 7.2, corresponde em adicionar três imagens em sequência com inter-



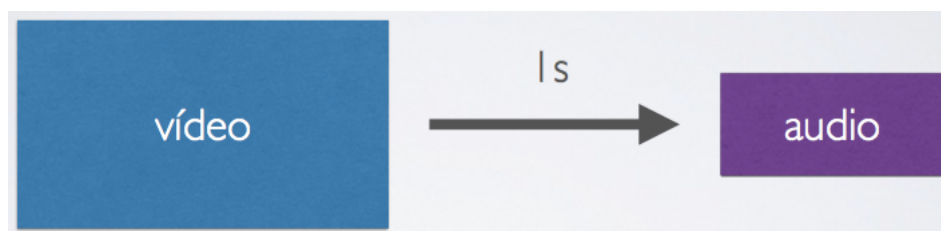


Figura 7.1: Etapa 1 da aplicação hipermídia realizada no teste de usabilidade

valo de 1s entre elas. Além disso, quando o áudio inicia, a primeira imagem da sequência começa sua apresentação.

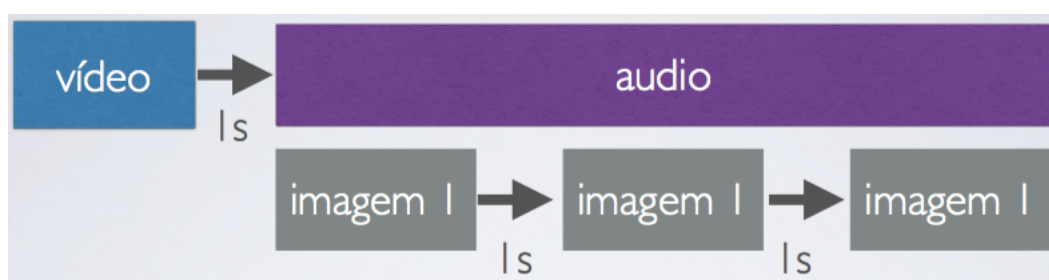


Figura 7.2: Etapa 2 da aplicação hipermídia realizada no teste de usabilidade

A terceira etapa do teste foca em utilizar a funcionalidade do STEVE de criar relações de interatividade. Para isso, foi necessário que os usuários adicionassem uma nova mídia representando um botão à aplicação que permitisse pular o vídeo de introdução. Nesta etapa, o botão deve iniciar e terminar juntamente com o vídeo e deve ser apresentado sobre o vídeo no canto inferior esquerdo da tela. Desta forma, o usuário necessitou editar algumas características de apresentação da mídia botão como inserir os valores 10% e 80% nos campos *Left* e *Top* respectivamente do painel *Position* e o valor 20% para os campos *Width* e *Height* do painel *Size*. Esses valores foram dados para os usuários, a fim de que eles pudessem focar na visão temporal da aplicação. Além disso, para tornar o botão interativo, os usuários tiveram que definir uma relação de interatividade envolvendo a mídia botão e o vídeo. Ela deveria utilizar o botão ENTER como tecla de interação com a mídia botão e finalizar a própria mídia botão e o vídeo após a interação do espectador. A visão temporal da aplicação desta última etapa é mostrada na Figura 7.3.

O teste foi realizado de forma independente pelos usuários, ou seja, os conceitos sobre apresentações multimídia, a introdução do STEVE e a descrição das etapas da aplicação hipermídia a ser criada no teste, foram dados aos usuários através de slides enviados por email a cada um dos usuários. Além disso, orientações sobre baixar e executar o STEVE foram dadas nos slides. Assim, em nenhum momento, os usuários puderam pedir ajuda técnica durante a realização dos testes. A única orientação sobre o STEVE estava

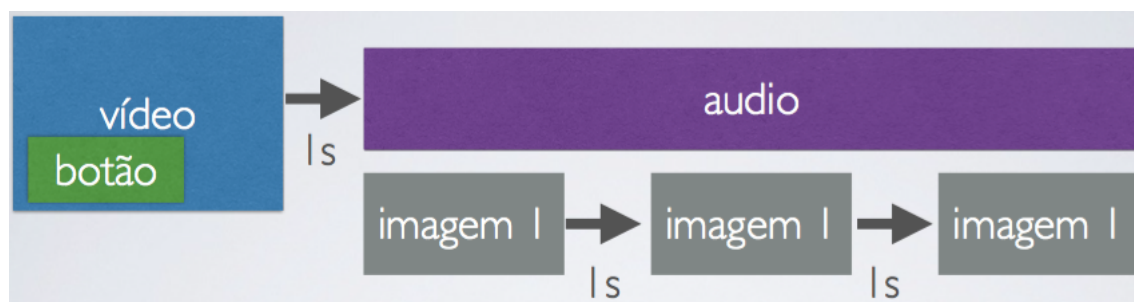


Figura 7.3: Etapa 3 da aplicação hipermídia realizada no teste de usabilidade

disponível nos slides. O questionário respondido pelos usuários ficou disponível online na plataforma Typeform [26].

### 7.2.2 Questionários

Para os usuários avaliarem o STEVE após a criação da aplicação hipermídia descrita na Subseção 7.2.1, eles foram submetidos a dois questionários. O primeiro utiliza perguntas definidas no SUS (System Usability Scale) [51] para medir o nível de eficácia e eficiência da ferramenta, e a satisfação do usuário. Neste método, as respostas usam uma escala de 1 (discordo totalmente) a 5 (concordo totalmente). A pontuação final está dentro do intervalo de 0 a 100. Quanto mais alto o valor, melhor foi a usabilidade percebida pelo usuário. Como as perguntas do SUS são muito gerais, elas foram editadas para o STEVE como pode ser visto abaixo.

1. Eu acho que gostaria de usar o STEVE com frequência;
2. Eu acho o STEVE desnecessariamente complexo;
3. Eu achei o STEVE fácil de usar;
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o STEVE;
5. Eu acho que várias funções do STEVE estão muito bem integradas;
6. Eu acho que o STEVE apresenta muita inconsistência;
7. Eu imagino que as pessoas aprenderão como usar o STEVE rapidamente;
8. Eu achei o STEVE confuso de usar;
9. Eu me senti confiante ao usar o STEVE;

10. Eu precisei aprender várias coisas novas antes de conseguir usar o STEVE.

Já o segundo questionário apresentou 12 perguntas com o objetivo de avaliar a expressividade dos componentes do STEVE e também sua interface gráfica. Além de coletar opiniões gerais sobre o uso do STEVE e de verificar se todas as etapas do teste de construção da aplicação hipermídia foram concluídas com sucesso e se os usuários já tinham utilizados algum editor de vídeo anteriormente. As 12 perguntas do segundo questionário estão listadas abaixo e suas respostas também variam de 1 (discordo totalmente) a 5 (concordo totalmente). Vale ressaltar que estas afirmações do segundo questionário foram definidas especificamente para este trabalho.

1. Eu achei a interface gráfica do STEVE simples e organizada;
2. Eu achei o estilo da interface gráfica do STEVE agradável;
3. Eu consegui importar facilmente arquivos para o repositório de mídias;
4. Eu achei fácil a adição de mídias na linha do tempo através do repositório de mídias;
5. Eu achei os ícones das relações temporais intuitivos;
6. Eu achei fácil sincronizar as mídias com as relações temporais oferecidas por STEVE;
7. Eu achei fácil definir uma mídia interativa;
8. Eu consegui entender facilmente a representação visual da interatividade;
9. Eu achei fácil o preenchimento das características de apresentação das mídias;
10. Eu consegui facilmente exportar minha aplicação para um documento HTML5;
11. Eu consegui executar com facilidade o documento HTML5 gerado por STEVE no navegador Web;
12. O uso do STEVE, em geral, foi agradável.

### 7.2.3 Resultados

Dos 10 usuários que realizaram o teste, apenas 2 não conseguiram finalizar todas as etapas do teste. Neste caso, a etapa 3 não foi concluída por esses dois. Vale ressaltar que um deles nunca tinha utilizado um editor de vídeo anteriormente. Os demais usuários conseguiram completar as 3 etapas do teste e executá-las no navegador com sucesso.

Em relação ao questionário SUS, foi obtida uma média de pontos SUS de 71,75 com desvio padrão de 11,55. Em uma escala de 0 (pouco satisfeito) a 100 (muito satisfeito), STEVE obteve uma usabilidade aceitável (70-100 pontos SUS) [29] e, a partir das questões, foi possível coletar percepções do usuário em relação a sua experiência ao utilizar a ferramenta. E pelos testes realizados, os usuários consideraram fácil o uso do STEVE. O gráfico com a pontuação obtida para cada usuário pode ser visualizado na Figura 7.4. Neste gráfico, apenas os usuários 3 e 4 conheciam a linguagem NCL, e os usuários 1, 3 e 7 nunca tinham utilizado alguma ferramenta de edição de vídeo.

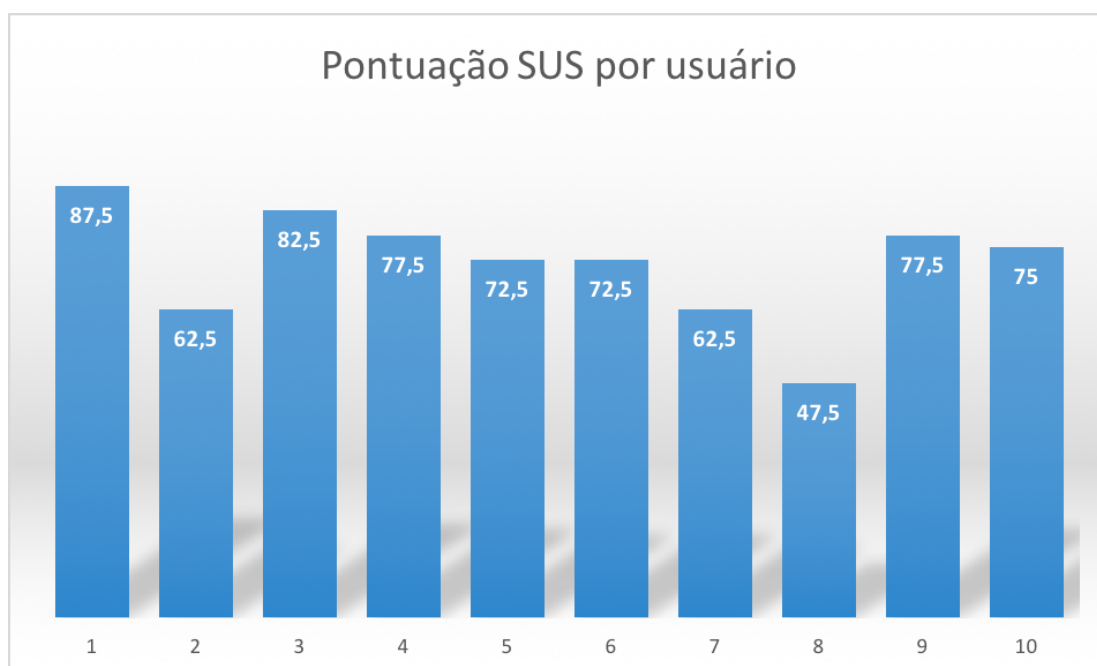


Figura 7.4: Resultado do questionário SUS

Para o segundo questionário, foi calculada a média para cada pergunta. Assim, uma média alta para uma determinada questão caracteriza que, em geral, os usuários concordaram com a afirmação descrita nela. A numeração das perguntas no eixo horizontal do gráfico 7.5 corresponde à ordem da lista do segundo questionário apresentado na Subseção 7.2.2. Neste gráfico, podemos notar que a última questão "O uso do STEVE, em geral, foi agradável." obteve uma média acima de 4, o que representa que os usuários tiveram uma boa experiência ao utilizar o STEVE. A alta média das questões 1 e 2 significa que o layout e estilo da interface gráfica do usuário agradaram muito bem os autores. Além disso, a pontuação elevada também das questões 10 e 11 mostra que o usuário facilmente exportou sua aplicação hipermídia criada no STEVE para o formato HTML5 e, em seguida, conseguiu executá-la com sucesso em um navegador Web.

As questões que obtiveram as menores médias foram a 3, 5, 6, 7 e 9. Isto indica que

a importação dos arquivos de mídia para o repositório do STEVE não está tão intuitiva, visto que também este ponto foi mencionado em diversos *feedbacks* no segundo questionário. Já a média obtida nas questões 5 e 6 sugere que os ícones das relações temporais e a definição propriamente dita podem ainda melhorar suas representações de forma a deixar mais intuitivo o comportamento temporal das relações oferecidas pela ferramenta. A média da questão 9 indica que a maneira de relacionar as propriedades de apresentação ao objeto de mídia atualmente selecionado na visão temporal também pode ser melhorada.

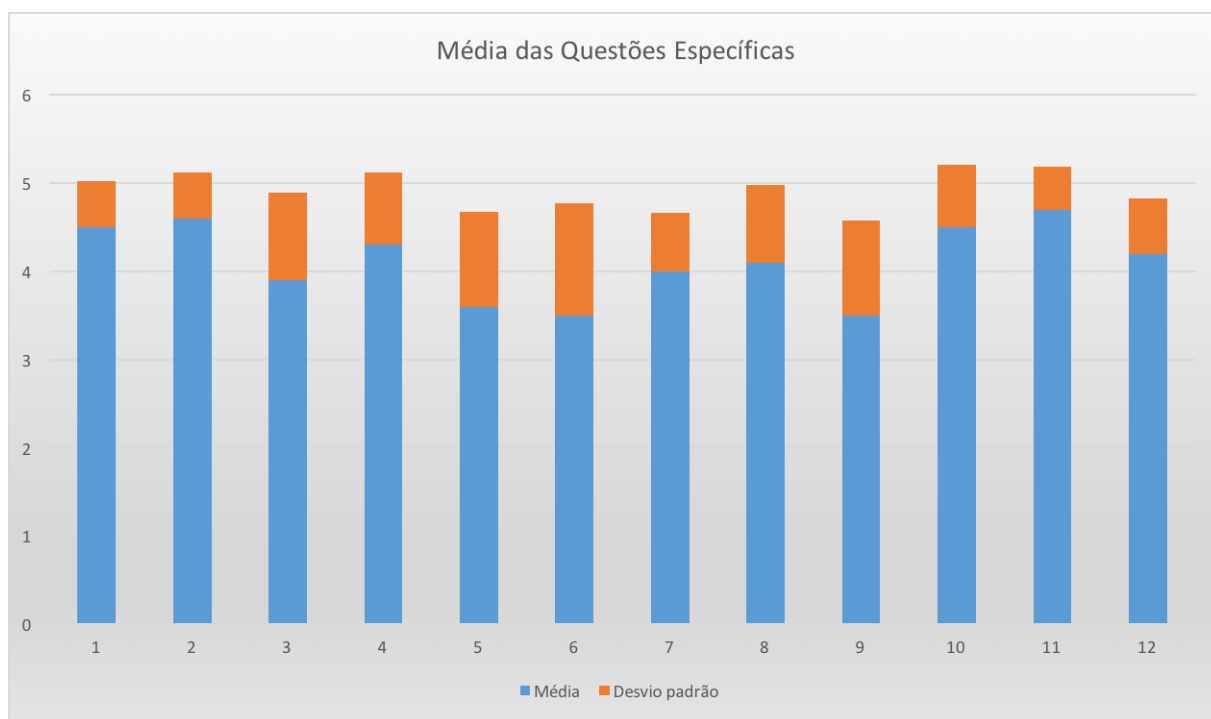


Figura 7.5: Pontuação média do questionário específico

Além disso, opiniões gerais foram coletadas sobre o uso do STEVE através de um campo textual livre no questionário. A primeira delas, como já mencionado, é em relação à importação de arquivos para o repositório de mídias do STEVE. Desta forma, a troca do botão de importação e a adição dessa opção na barra de menu da ferramenta poderia facilitar este processo. Outro ponto mencionado foi em relação aos alertas exibidos no momento da criação das relações temporais. Assim, a reformulação desses alertas poderia melhorar o uso das relações oferecidas por STEVE. Ainda foram levantadas questões sobre o uso do botão ENTER para confirmar nas janelas exibidas pela ferramenta e também a função de refazer ações.

## 7.3 Conclusão

Este capítulo apresentou uma discussão dos trabalhos relacionados, enfatizando os pontos fortes e fracos de cada um deles. Discutiu também o teste de usabilidade realizado para avaliar a ferramenta STEVE. Além disso, a metodologia utilizada no teste foi discutida e os questionários para avaliar a eficácia, eficiência, satisfação do usuário e as funcionalidades da ferramenta foram apresentados. Ao final, os resultados obtidos do teste foram discutidos, os quais permitiram coletar percepções dos usuários em relação às funcionalidades citadas nos questionários. A partir dos testes realizados pelos usuários, eles consideraram fácil o uso da ferramenta STEVE. Ademais, alguns pontos a melhorar também foram levantados.

O próximo capítulo irá apontar as contribuições desta dissertação, suas limitações e trabalhos futuros.

# Capítulo 8

## Conclusão

Tendo como motivação evitar as limitações intrínsecas do paradigma *timeline* utilizado, principalmente, pelas ferramentas comerciais e facilitar a autoria de aplicações hipermídia, esta dissertação de mestrado realizou a proposta de um modelo de documento hipermídia, denominado SIMM (Simple Interactive Multimedia Model). Este modelo faz uso da representação de sincronização temporal baseada em eventos, especificando os relacionamentos temporais e espaciais entre os itens de mídia de acordo com os eventos ocorridos na aplicação. Este modelo é uma simplificação do modelo NCM, a fim de facilitar a autoria de documentos hipermídia quando utilizado em uma implementação de um editor gráfico baseado em visão temporal.

Visto que as ferramentas acadêmicas que fazem uso da sincronização temporal baseada em eventos exigem que o autor tenha conhecimentos sobre o modelo utilizado, esta dissertação propôs um editor gráfico da visão espaço-temporal, chamado de STEVE (Spatial-Temporal View Editor), baseado no modelo SIMM, visando permitir que autores sem conhecimento em programação ou em alguma linguagem padrão de autoria multimídia possam criar suas próprias aplicações hipermídia para diversas plataformas.

Este capítulo final discute a contribuição do trabalho na Seção 8.1, apresentando as funcionalidades da ferramenta proposta. Em seguida, as limitações do trabalho são discutidas na Seção 8.2. E na Seção 8.3, os trabalhos futuros são apresentados.

### 8.1 Contribuição

O modelo de documento hipermídia chamado SIMM (Simple Interactive Multimedia Model) e o editor gráfico da visão espaço-temporal denominado STEVE (Spatio-Temporal

View Editor), que oferece a definição de relações temporais causais entre os itens de mídia e de interatividade, são as principais contribuições desta dissertação de mestrado. Elas permitem e facilitam a autoria de documentos hipermídia mesmo por usuários sem conhecimento em programação ou em linguagens de autoria multimídia.

SIMM utiliza o modelo de sincronização temporal baseado em eventos, sendo expressivo na definição dos relacionamentos temporais e oferecendo suporte à definição da interatividade. O modelo SIMM é baseado nas entidades básicas do modelo NCM, nós e elos, e nos seus tipos de eventos: de apresentação e de seleção. Ele define um conjunto de entidades simplificado relativo à hierarquia de classes presentes no modelo NCM. Desta forma, SIMM apresenta um modelo de sincronização temporal com um nível de abstração relativamente mais alto do que o NCM.

No modelo SIMM, as relações temporais são definidas apenas utilizando a especificação de uma mídia mestre, de uma lista de escravos, do tipo da relação e, opcionalmente, do valor de atraso. Esta simplificação do SIMM fornece um modelo que pode ser usado para implementação de um sistema de autoria hipermídia que ofereça relações temporais simples, práticas e razoavelmente expressivas para representar as aplicações hipermídia. Além de oferecer entidades que realizam a sincronização temporal do documento, SIMM também possui entidades que implementam a sincronização espacial dos itens de mídias. E ainda apresenta classes que representam as características de apresentação dos objetos de mídia.

STEVE permite que autores sem conhecimento em programação sejam capazes de criar apresentações multimídia interativas utilizando a visão espaço-temporal. Ele também permite definir eventos de interatividade e relações temporais causais entre os itens de mídia. A ferramenta evita as limitações do paradigma *timeline* ao fazer uso do modelo de sincronização temporal baseado em eventos. As relações temporais causais oferecidas por STEVE através da barra de botões da visão temporal foram baseadas nas 13 relações temporais entre intervalos de Allen [27]. Dessa forma, a ferramenta permite que usuários leigos consigam criar relações temporais causais entre os itens de mídia sem ter conhecimento sobre modelos hipermídia que utilizam a sincronização temporal baseada em eventos.

Durante a edição das relações temporais, STEVE realiza o tratamento das ações do usuário, de forma a manter a consistência temporal da aplicação, assim, garantindo que as aplicações hipermídia se comportem conforme a sincronização temporal estabelecida na fase de autoria.



Além disso, STEVE oferece painéis de propriedade em sua interface gráfica para permitir que os usuários possam editar as principais características de apresentação, como tamanho, posição e transparência, dos objetos de mídia. A ferramenta também disponibiliza ao usuário a prévia das aplicações hipermídia, possibilitando que os autores visualizem como e onde as mídias serão apresentadas durante a execução da aplicação.

A definição de interatividade também é suportada por STEVE, onde o autor apenas seleciona a mídia que deseja tornar interativa e informa a tecla/botão para interação e as ações que irão ocorrer após o espectador interagir com a aplicação. Para que o autor possa verificar o comportamento da aplicação caso a interatividade ocorresse, STEVE permite que o autor simule eventos interativos através de instantes hipotéticos para a ocorrência desses eventos.

Após o término da criação da aplicação no STEVE, a ferramenta permite que os autores exportem suas aplicações para documentos NCL e HTML 5, fazendo com que seja possível executá-las em diversas plataformas.

## 8.2 Limitações do Editor STEVE

As limitações do modelo SIMM foram discutidas na Seção 4.7. Visto que a ferramenta STEVE foi implementada baseando-se neste modelo, consequentemente, STEVE as adquire. Além disso, STEVE possui outras limitações independentes do modelo SIMM, que são descritas a seguir.

A primeira delas consiste em não suportar a definição de âncoras de conteúdo. Estas âncoras permitem especificar segmentos do conteúdo das mídias, os quais podem participar de relações de sincronização temporal com outras âncoras ou mídias de um documento hipermídia. Não permitir a especificação de relações espaciais entre as mídias de uma aplicação hipermídia é outra limitação do editor STEVE. Além disso, a ferramenta não oferece a execução das aplicações hipermídia criadas no próprio editor. A implementação da prévia dos documentos criados no STEVE deve ser finalizada, o que representa uma limitação da ferramenta. Outra limitação é encontrada na importação de documentos NCL para dentro do editor STEVE. Na importação NCL, os documentos hipermídia perdem as definições das relações temporais dos itens de mídia, ou seja, nenhuma relação é mapeada para as relações definidas pelo modelo SIMM.

## 8.3 Trabalhos Futuros

Como trabalho futuro, as sugestões dadas no teste de usabilidade realizado devem ser analisadas e implementadas, a fim de melhorar a usabilidade do STEVE. Além disso, novos testes de usabilidade devem ser realizados para avaliar com maior foco a visão espacial e novamente a visão temporal, após a implementação das melhorias sugeridas no primeiro teste. Ademais, mais testes de sistemas devem ser realizados para verificar possíveis inconsistências.

Outras funcionalidades, que são independentes dos recursos oferecidos pelo modelo SIMM, podem ser adicionadas à ferramenta STEVE a fim de enriquecer seu poder de edição. O primeiro deles é estender o suporte da ferramenta STEVE a outras linguagens de autoria multimídia, por exemplo, SMIL [7]. Visto que diversas cadeias temporais podem compor a visão temporal de uma aplicação hipermídia, é interessante disponibilizar aos autores uma visão em grafo relacionando cada uma das cadeias temporais. Outra funcionalidade que facilitaria a edição das características de apresentação é permitir que os usuários as editem diretamente na visão espacial da aplicação, ou seja, onde é visualizada a prévia da aplicação. A importação de documentos NCL para dentro do STEVE remove todos os elos temporais do documento. Assim, o mapeamento desses elos para as relações temporais do modelo SIMM utilizado por STEVE enriqueceria a funcionalidade de importação de documentos NCL para a ferramenta. Outro recurso também relevante é implementar a validação espacial da aplicação.

Alguns suportes oferecidos pelo modelo SIMM não foram implementados pela ferramenta STEVE como as âncoras e as relações espaciais. Dessa forma, implementar estes recursos enriqueceria a edição da visão espacial das aplicações e a especificação da sincronização temporal na ferramenta.

Outros recursos, que dependem da extensão do modelo SIMM, são os seguintes. Permitir alterar propriedades de apresentação dos objetos de mídia através das relações de interatividade tornaria ainda mais interessante a funcionalidade de definição de mídias interativas no STEVE. Para isso, o modelo SIMM deve suportar testes de variáveis. Definir animações aumentaria o poder de edição da visão espacial do STEVE. E a inclusão de templates na ferramenta facilitaria ainda mais a autoria de aplicações hipermídia por usuários sem conhecimento em linguagens ou modelos multimídia. Além disso, definir novas abstrações para o modelo SIMM, a fim de simplificar ainda mais a definição de aplicações hipermídia.

# Referências

- [1] Folha de estilo ncl4web.xsl. <http://www.midiacom.uff.br/cableb/ncl4web/ncl4web.xsl>. Último acesso: 2016.
- [2] Package javax.xml.transform. <https://docs.oracle.com/javase/7/docs/api/javax/xml/transform/summary.html>. Último acesso: 2016.
- [3] Package org.w3c.dom. <https://docs.oracle.com/javase/7/docs/api/org/w3c/dom/package-summary.html>. Último acesso: 2016.
- [4] Ergonomic requirements for office work with visual display terminals (vdts) part 11: guidance on usability, 1998. ISO Standard 9241-11:1998.
- [5] Macromedia authorware. <http://www.adobe.com/products/authorware/>, 2003.
- [6] *Digital terrestrial television - Data coding and transmission specification for digital broadcasting Part 2: Ginga-NCL for fixed and mobile receivers - XML application language for application coding*. ABNT 15606-2, 2007.
- [7] Synchronized Multimedia Integration Language - SMIL. <http://www.w3.org/TR/SMIL/>, 2008.
- [8] *Nested Context Language (NCL) and Ginga-NCL for IPTV*. ITU H.761 recommendation, 2009.
- [9] Adobe Flash. <http://www.adobe.com/devnet/actionscript/references.html>., 2010.
- [10] Construindo programas audiovisuais interativos utilizando a ncl 3.0. <http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/TutorialNCL3.0-2ed-3rev.pdf>, 2010.
- [11] ISO 9241-210:2010. Ergonomics of human-system interaction - part 210: Human-centred design for interactive systems. [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=52075](http://www.iso.org/iso/catalogue_detail.htm?csnumber=52075), 2010.
- [12] Obrigatoriedade da implementação do ginga em tvs digitais. <http://forumsbtvd.org.br/obrigacao-da-incorporacao-do-ginga-e-estendida-para-tvs-de-plasma/>, 2012. Último acesso: março de 2016.
- [13] Google i/o 2014 - material design: Structure and components. <https://www.youtube.com/watch?v=dZqzz5lZFvo>, 2014.
- [14] Html5. <http://www.w3.org/TR/html5/>, 2014.
- [15] Java fx 8. <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784>, 2014.

- [16] Javafx: Working with javafx ui components. <http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/bar-chart.htm#CIHJFHDE>, 2014.
- [17] Pesquisa brasileira de mídia 2015. secretaria de comunicação social da presidência da república. <http://www.secom.gov.br/atuacao/pesquisa/lista-de-pesquisas-quantitativas-e-qualitativas-de-contratos-atuais/pesquisa-brasileira-de-midia-pbm-2015.pdf/view>, 2015.
- [18] Adobe Premiere. <http://www.adobe.com/br/products/premiere.html>, 2016.
- [19] Davinci Resolve 12. <https://www.blackmagicdesign.com/products/davinciresolve>, 2016.
- [20] Final Cut Pro X. <http://www.apple.com/br/final-cut-pro/>, 2016.
- [21] iMovie versão 10.1. <http://www.apple.com/br/mac/imovie/>, 2016.
- [22] Limsee 2: un editor visuale opensource. <http://www.html.it/pag/16863/limsee-2-un-editor-visuale-opensource/>, 2016.
- [23] Material design. <https://www.google.com/design/spec/material-design/introduction.html>, 2016.
- [24] Material design icons. <https://design.google.com/icons>, 2016.
- [25] Nero Video 2016. <http://www.nero.com/ptb/products/nero-video/>, 2016.
- [26] Typeform. <https://www.typeform.com/>, 2016.
- [27] ALLEN, J. F. Maintaining knowledge about temporal intervals. *Commun. ACM* 26, 11 (1983), 832–843.
- [28] BAILEY, B.; KONSTAN, J. A.; COOLEY, R.; DEJONG, M. Readings in multimedia computing and networking. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, ch. Nsync: A Toolkit for Building Interactive Multimedia Presentations, pp. 761–770.
- [29] BANGOR, A.; KORTUM, P.; MILLER, J. Determining what individual sus scores mean: Adding an adjective rating scale. *J. Usability Studies* 4, 3 (May 2009), 114–123.
- [30] BEVAN, N. What is the difference between the purpose of usability and user experience evaluation methods. In *Proceedings of the Workshop UXEM 2009* (2009).
- [31] BEVAN, N.; CARTER, J.; HARKER, S. What have we learnt about usability since 1998? In *Human-Computer Interaction: Design and Evaluation*, M. Kurosu, Ed., vol. 9169 of *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 143–151.
- [32] BLAKOWSKI, G.; STEINMETZ, R. A media synchronization survey: reference model, specification, and case studies. *Selected Areas in Communications, IEEE Journal on* 14, 1 (Jan 1996), 5–35.

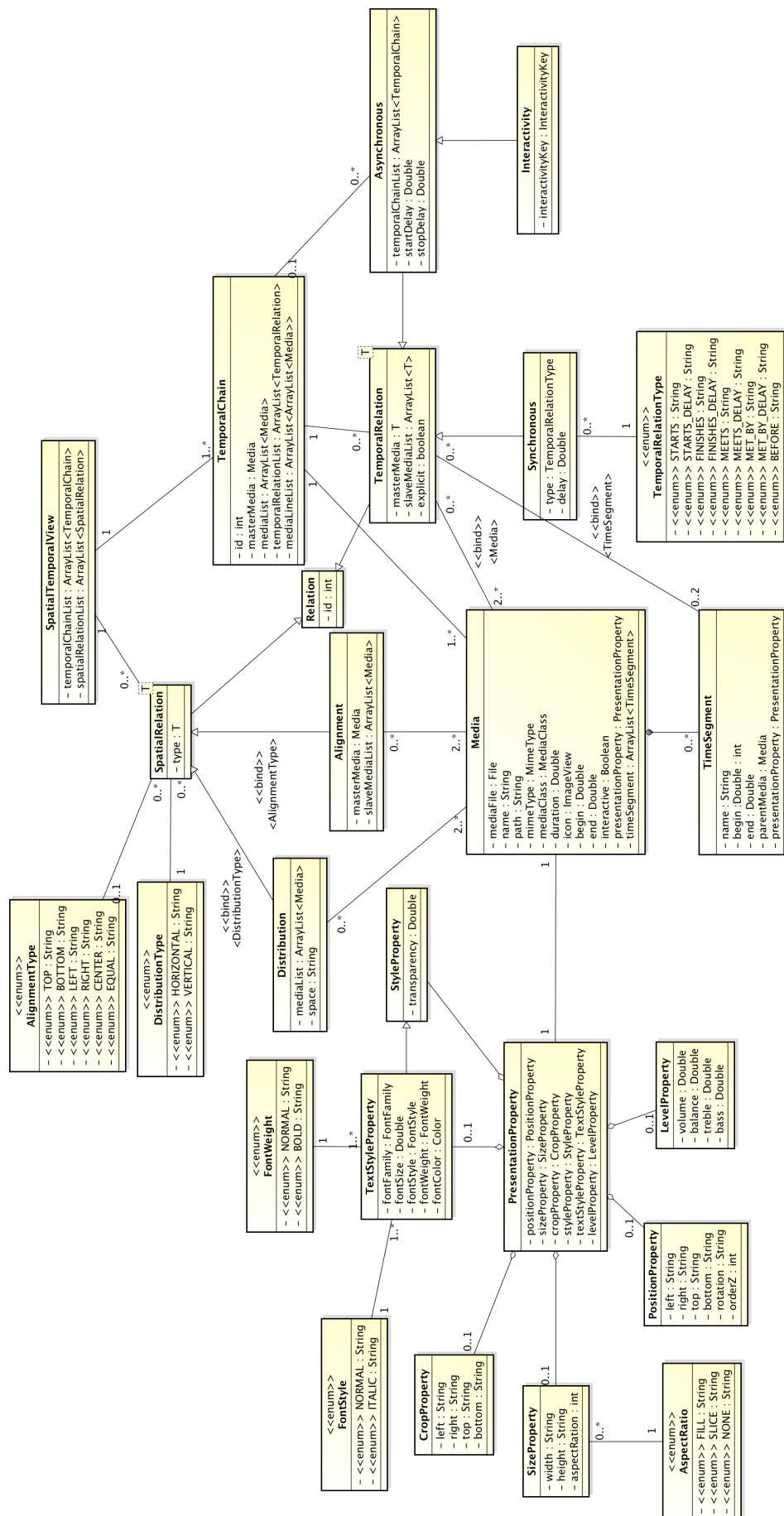
- [33] BUCHANAN, M.; ZELLWEGER, P. Specifying Temporal Behavior in Hypermedia Documents. In *Proceedings of European Conference on Hypertext* (1992).
- [34] BUCHANAN, M. C.; ZELLWEGER, P. T. Automatic temporal layout mechanisms revisited. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1 (Feb. 2005), 60–88.
- [35] BULTERMAN, D.; HARDMEN, L.; JASEN, J.; MULLENDER, K.; RUTLEDGE, L. GRiNS: A GRaphical INTERface for creating and playing SMIL documents. *WWW7 Conference, Computer Networks and ISDN Systems* 30, 1-7 (1998), 519–529.
- [36] BULTERMAN, D. C. A.; HARDMAN, L. Structured multimedia authoring. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1 (2005), 89–109.
- [37] CERQUEIRA NETO, J. R.; MESQUITA SANTOS, R. C.; SOARES NETO, C. S.; TEIXEIRA, M. M. Eventline: Representation of the temporal behavior of multimedia applications. In *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web* (São Paulo, Brazil, 2012), WebMedia '12, ACM, pp. 215–222.
- [38] COSTA, R. M. R. *Modelling and Authoring Hypermedia Documents*. Ph.d. thesis, Univ. Amsterdam, 1998.
- [39] COSTA, R. M. R. *Controle do Sincronismo Temporal de Aplicações Hipermédia*. Tese de doutorado, Pontifícia Universidade Católica do Rio de Janeiro, 2010.
- [40] COSTA, R. M. R.; SOARES, L. F. G. Modelos Temporais para Especificação de Sincronismo em Documentos Hipermédia. Monografia, Pontifícia Universidade Católica do Rio de Janeiro, 2005.
- [41] DELTOUR R, LAYAIDA N, W. D. Limsee2: a cross-platform smil authoring tool. In *the European research consortium for informatics and mathematics - ERCIM News* (2005), p. 61.
- [42] DOS SANTOS, J. A.; BRAGA, C.; MUCHALUAT-SAADE, D. C. Automating the analysis of ncl documents with a model-driven approach. In *Proceedings of the 19th Brazilian Symposium on Multimedia and the Web* (New York, NY, USA, 2013), WebMedia '13, pp. 193–200.
- [43] DOS SANTOS, J. A.; BRAGA, C.; MUCHALUAT-SAADE, D. C.; ROISIN, C.; LAYAIDA, N. Spatio-temporal validation of multimedia documents. In *Proceedings of the 2015 ACM Symposium on Document Engineering* (Lausanne, Switzerland, 2015), DocEng '15, ACM, pp. 133–142.
- [44] FUJIKAWA, K., S. S. M. T. N. S.; MIYAHARA, H. Multimedia presentation system harmony with temporal and active media. In *In: Proc. of the Summer 1991 USENIX Conference, June* (Nashville, TN,, 1991), pp. 75–93.
- [45] GUIMARÃES, R. L. Composeur um ambiente de autoria de documentos NCL para TV digital interativa. Dissertação de mestrado, Pontifícia Universidade Católica do Rio de Janeiro, 2007.
- [46] GUIMARÃES, R. L.; COSTA, R.; SOARES, L. F. G. Composeur: Authoring Tool for iTV Programs. *European Interactive TV Conference - EuroITV* (2008).

- [47] HALASZ, F.; SCHWARTZ, M. The Dexter Hypertext Reference Model. *NIST Hypertext Standardization Workshop* (1990).
- [48] HARDMAN, L.; BULTERMAN, D. C. A. Authoring support for durable interactive multimedia presentations. In *Aug - 1 Sep* (1995), pp. 119–143.
- [49] HARDMAN, L.; VAN ROSSUM, G.; BULTERMAN, D. The Amsterdam Hypermedia Model: extending hypertext to support real multimedia. *Hypermedia Journal* 5, 1 (1993), 47–69.
- [50] IERUSALIMSKY, R.; DE FIGUEIREDO, L. H.; FILHO, W. C. Lua—an extensible extension language. *Softw. Pract. Exper.* 26, 6 (1996), 635–652.
- [51] J, B. Sus: a quick and dirty usability scale. In: *Jordan P, Thomas B, Weerdmeester B, McClelland I (eds) Usability evaluation in industry. Taylor & Francis* (1996), 189–194.
- [52] LAW, E. L.-C.; ROTO, V.; HASSENZAH, M.; VERMEEREN, A. P.; KORT, J. Understanding, scoping and defining user experience: A survey approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI '09, ACM, pp. 719–728.
- [53] LIMA, B.; AZEVEDO, R.; MORENO, M.; SOARES, L. F. G. Composer 3: Ambiente de Autoria Extensível, Adaptável e Multiplataforma. *WebMedia - Workshop de TV Digital Interativa (WTVDI)* (2010).
- [54] MATTOS, D.P. E SAADE, D. Exibição da visão temporal de aplicações NCL para TV digital. Trabalho de conclusão de curso, Universidade Federal Fluminense, 2013.
- [55] MATTOS, D. P.; SILVA, J. V.; MUCHALUAT-SAADE, D. NEXT: Graphical editor for authoring NCL documents supporting composite templates. In *Proceedings of the 11th European Conference on Interactive TV and Video* (New York, NY, USA, 2013), EuroITV '13, ACM, pp. 89–98.
- [56] MUCHALUAT-SAADE, D. C.; SOARES, L. F. G. XConnector & XTemplate: Improving the Expressiveness and Reuse in Web Authoring Languages. *New Review of Hypermedia and Multimedia* 8 (2002), 139–169.
- [57] OGAWA, R., H. H.; KANEKO, A. Scenario-based hypermedia: A model and a system. In *ECHT 90 Nov.* (INRIA France, 1990), pp. 38–51.
- [58] PÉREZ-LUQUE, M.; LITTLE, T. A Temporal Reference Framework for Multimedia Synchronization. *IEEE Journal on Selected Areas in Communications* 14, 1 (1996), 36–51.
- [59] SANTOS, J. A. F.; MUCHALUAT-SAADE, D. C. XTemplate 3.0: Spatio-temporal Semantics and Structure Reuse for Hypermedia Compositions. *Multimedia Tools and Applications* 61, 3 (2012), 645–673.
- [60] SANTOS, J. A. F.; SILVA, J. V.; VASCONCELOS, R.; SCHAU, W.; WERNER, C.; MUCHALUAT-SAADE, D. C. aNa: API for NCL Authoring. *WebMedia - Workshop de Ferramentas e Aplicações* (2012).

- [61] SILVA, E. C. O.; SANTOS, J. A. F.; MUCHALUAT-SAADE, D. C. NCL4WEB: Translating NCL applications to HTML5 web pages. In *Proceedings of the 2013 ACM Symposium on Document Engineering* (New York, NY, USA, 2013), DocEng '13, pp. 253–262.
- [62] SOARES, L. F. G.; BARBOSA, S. D. J. *Programando em NCL 3.0*, 2 ed. TeleMídia, 2012.
- [63] SOARES, L. F. G.; BARBOSA, S. D. J. *Programando em NCL 3.0, Apêndice G - HTG*, 2 ed. TeleMídia, 2012.
- [64] SOARES, L. F. G.; RODRIGUES, R. *Nested Context Language 3.0 Part 8 NCL Digital TV Profiles. MCC 35/06 Technical Report*. ABNT 15606-2, 2006.
- [65] SOARES, L. F. G.; RODRIGUES, R.; MUCHALUAT-SAADE, D. C. Modeling Authoring and Formatting Hypermedia Documents in The Hyperprop System. *ACM Multimedia Systems Journal* 8, 2 (2000), 118–134.
- [66] SOARES, L. F. G.; RODRIGUES, R. F.; MUCHALUAT-SAADE, D. C. Modeling, Authoring and Formatting Hypermedia Documents in the HyperProp System. *Multimedia Systems archive* 8 (2000), 118 – 134.
- [67] TIEN, T.; ROISIN, C. A multimedia model based on structured media and sub-elements for complex multimedia authoring and presentation. *International Journal of Software Engineering and Knowledge Engineering* 12, 05 (2002), 473–500.
- [68] VAN ROSSUM, G.; JANSEN, J.; MULLENDER, K. S.; BULTERMAN, D. CMIFed: A Presentation Environment for Portable Hypermedia Documents. In *ACM Multimedia* (1993).
- [69] WILLRICH, R.; SAQUI-SANNES, P.; S'NAC, P.; DIAZ, M. *Design and management of multimedia information systems*. In: RAHMAN, S. M. (Ed.) [S.l.]: IGI Publishing, 2001.

## APÊNDICE A - Diagrama de Classe Completo do Modelo do STEVE





## APÊNDICE B - A Linguagem NCL

A linguagem NCL (*Nested Context Language*) [64] é baseada no modelo NCM [65] e na linguagem XML, ou seja, ela usa elementos XML para definir as entidades do modelo NCM para especificar um documento hipermídia.

O documento NCL possui, basicamente, duas partes: cabeçalho (*head*) e o corpo (*body*). A primeira contém as bases de região (*regionBase*), que possuem as regiões da tela onde as mídias são apresentadas, a base de descritores (*descriptorBase*) que define características de apresentação das mídias, e as bases de conectores (*connectorBase*) que especificam os conectores utilizados pelos elos do documento. A seguir, é mostrado o cabeçalho de um documento NCL.

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <ncl id="exemplo" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
3   <head>
4     <regionBase>
5       <region id="backgroundReg" width="100%" height="100%"
6         zIndex="1"/>
7       (...)
8     </regionBase>
9     <descriptorBase>
10      <descriptor id="backgroundDesc" region="backgroundReg"/>
11      (...)
12    </descriptorBase>
13    <connectorBase>
14      <causalConnector id="onBeginStart">
15        <simpleCondition role="onBegin" max="unbounded"
16          qualifier="and"/>
17        <simpleAction role="start" max="unbounded"
18          qualifier="par"/>
19      </causalConnector>
20    </connectorBase>
21  </head>

```

22     (...)  
 23     </ncl>

O corpo contém os nós de mídia (*media*), contextos (*context* e *switch*), no mínimo uma porta (*port*) indicando o início da apresentação do documento e os elos (*link*).

Para os nós podem ser definidas interfaces, a fim de criar relacionamentos entre eles, as quais são: *area* que define uma âncora de conteúdo (trecho de mídia), *port* que mapeia uma interface ou componente do contexto, *property* que define uma âncora de propriedade e *switchPort* que mapeia uma interface dos componentes de um *switch* (contexto com nós alternativos, onde apenas um deles é ativado na apresentação do documento hipermídia).

O elemento *link* possui elementos denominados *binds*, cujos atributos principais são *role* e *component*, que realizam a ligação dos nós com seus papéis definidos na semântica da relação causal do conector usado pela relação. A especificação do conector a ser usado no elo é indicado pelo atributo *xconnector*. Os papéis padrões, tanto para condição quanto ação, são mostrados na Tabela B.1.

Valor do <i>role</i>	Valor da Transição	Tipo do evento
<i>onBegin</i>	<i>starts</i>	<i>presentation</i>
<i>onEnd</i>	<i>stops</i>	<i>presentation</i>
<i>onAbort</i>	<i>aborts</i>	<i>presentation</i>
<i>onPause</i>	<i>pauses</i>	<i>presentation</i>
<i>onResume</i>	<i>resumes</i>	<i>presentation</i>
<i>onSelection</i>	<i>starts</i>	<i>selection</i>
<i>onBeginAttribution</i>	<i>starts</i>	<i>attribution</i>
<i>onEndAttribution</i>	<i>stops</i>	<i>attribution</i>

(a) Papéis de Condição

Valor do <i>role</i>	Valor do Tipo de Ação	Tipo do evento
<i>start</i>	<i>start</i>	<i>presentation</i>
<i>stop</i>	<i>stop</i>	<i>presentation</i>
<i>abort</i>	<i>abort</i>	<i>presentation</i>
<i>pause</i>	<i>pause</i>	<i>presentation</i>
<i>resume</i>	<i>resume</i>	<i>presentation</i>
<i>set</i>	<i>start</i>	<i>attribution</i>

(b) Papéis de Ação

Tabela B.1: Papéis Padrões

Além destas condições predefinidas, é possível construir outras que testem o estado de apresentação de um evento, atributos associados a eventos ou valores de propriedade. Isto é feito através do elemento *assessmentStatement*. Ele possui ainda os elementos *attribu-*

*teAssessment* que define um atributo cujo valor é testado e o elemento *valueAssessment* que define o valor.

Na definição de conectores, as condições simples e compostas são representadas pelos elementos *simpleCondition* e *compoundCondition* respectivamente. Este último contém uma ou mais condições simples ou compostas e ainda pode possuir um ou mais elementos *assessmentStatement*. Já as ações simples e compostas são simbolizadas pelos elementos *simpleAction* e *compoundAction* nesta ordem. Sendo que uma ação composta pode conter uma ou mais ações simples ou compostas. A seguir é mostrado um documento NCL como exemplo, apresentando a estrutura básica e os principais elementos discutidos.

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <ncl id="exemplo" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
3   <head>
4     (...)
5   </head>
6
7   <body>
8     <port id="entry" component="animation"/>
9     <media id="background" src="background.png"
10    descriptor="backgroundDesc"/>
11    <media id="animation" src="animGar.mpg" descriptor="screenDesc">
12      <area id="segPhoto" begin="12s"/>
13      <area id="segIcon" begin="5s" end="10s"/>
14      <property name="bounds"/>
15    </media>
16    <media id="music" src="choro.mp3" descriptor="audioDesc"/>
17    <media id="photo" src="photo.png" descriptor="photoDesc"/>
18
19    <context id="advert">
20      <media id="reusedAnimation" refer="animation"
21      instance="instSame">
22        (...)
23        <link id="lBeginShoes"
24        xconnector="onKeySelectionStopSetStart">
25          <bind role="onSelection" component="icon">
26            <bindParam name="tecla" value="RED"/>
27          </bind>
28          <bind role="start" component="shoes"/>
29          <bind role="start" component="ptForm"/>
30          <bind role="set" component="reusedAnimation"
31          interface="bounds"/>

```

```
32         <bind role="stop" component="icon"/>
33     </link>
34     (...)
35 </context>
36
37 <link id="IMusic" xconnector="onBeginStart">
38     <bind role="onBegin" component="animation"/>
39     <bind role="start" component="music"/>
40 </link>
41     (...)
42 </body>
43 </ncl>
```

## APÊNDICE C - Base de Conectores

Este apêndice apresenta todos os conectores hiperímia [56] usados pelo editor STEVE que dão a semântica da relação causal aos elementos `<link>` da linguagem NCL.

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2
3 <ncl id="causalConnBase" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
4
5     <head>
6
7         <connectorBase id="causalConnectorBase">
8
9             <causalConnector id="onBeginStart">
10                 <simpleCondition role="onBegin"/>
11                 <simpleAction role="start" max="unbounded"/>
12             </causalConnector>
13
14             <causalConnector id="onBeginStartDelay">
15                 <connectorParam name="delay"/>
16                 <simpleCondition role="onBegin"/>
17                 <simpleAction role="start" delay="$delay"
18                     max="unbounded"/>
19             </causalConnector>
20
21             <causalConnector id="onEndStop">
22                 <simpleCondition role="onEnd"/>
23                 <simpleAction role="stop" max="unbounded"/>
24             </causalConnector>
25
26             <causalConnector id="onEndStopDelay">
27                 <connectorParam name="delay"/>
28                 <simpleCondition role="onEnd"/>
29                 <simpleAction role="stop" delay="$delay"
30                     max="unbounded"/>

```

```

31         </causalConnector>
32
33         <causalConnector id="onEndStart">
34             <simpleCondition role="onEnd"/>
35             <simpleAction role="start" max="unbounded"/>
36         </causalConnector>
37
38         <causalConnector id="onEndStartDelay">
39             <connectorParam name="delay"/>
40             <simpleCondition role="onEnd"/>
41             <simpleAction role="start" delay="$delay"
42                 max="unbounded"/>
43         </causalConnector>
44
45         <causalConnector id="onBeginStop">
46             <simpleCondition role="onBegin"/>
47             <simpleAction role="stop" max="unbounded"/>
48         </causalConnector>
49
50         <causalConnector id="onBeginStopDelay">
51             <connectorParam name="delay"/>
52             <simpleCondition role="onBegin"/>
53             <simpleAction role="stop" delay="$delay"
54                 max="unbounded"/>
55         </causalConnector>
56
57         <causalConnector id="onSelectionStartStop">
58             <simpleCondition role="onSelection"/>
59             <compoundAction operator="par">
60                 <simpleAction role="start" max="unbounded"/>
61                 <simpleAction role="stop" max="unbounded"/>
62             </compoundAction>
63         </causalConnector>
64
65         <causalConnector id="onSelectionStartStopDelay">
66             <connectorParam name="stopDelay"/>
67             <connectorParam name="startDelay"/>
68             <simpleCondition role="onSelection"/>
69             <compoundAction operator="par">
70                 <simpleAction role="stop" delay="$stopDelay"
71                     max="unbounded"/>
72                 <simpleAction role="start" delay="$startDelay"
73                     max="unbounded"/>

```

```
74         </compoundAction>
75     </causalConnector>
76
77     </connectorBase>
78
79 </head>
80
81 </ncl>
```