

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DA JANEIRO**

**FÁBIO RODRIGUES COSTA**

**UM EDITOR GRÁFICO PARA DEFINIÇÃO E  
EXIBIÇÃO DO SINCRONISMO DE DOCUMENTOS  
MULTIMÍDIA/HIPERMÍDIA**

**DISSERTAÇÃO DE MESTRADO**

**DEPARTAMENTO DE INFORMÁTICA**

Rio de Janeiro, 23 de agosto de 1996

**Fábio Rodrigues Costa**

**UM EDITOR GRÁFICO PARA DEFINIÇÃO E  
EXIBIÇÃO DO SINCRONISMO DE DOCUMENTOS  
MULTIMÍDIA/HIPERMÍDIA**

Dissertação apresentada ao Departamento  
de Informática da PUC-RJ como parte  
dos requisitos para obtenção do título de  
Mestre em Ciências em Informática.

Orientador: Luiz Fernando Gomes Soares

**Departamento de Informática**

**Pontifícia Universidade Católica do Rio de Janeiro**

**Rio de Janeiro, 23 de agosto de 1996**

**Este trabalho é dedicado**

**a Marlúcia, pelo amor, carinho e compreensão  
em mim depositados;**

**e aos meus pais José Faustino Costa e Luiza  
Rodrigues Costa, por todo amor, atenção e  
ensinamentos que já me dedicaram.**

## **AGRADECIMENTOS**

- ao meu Professor Orientador Luiz Fernando Gomes Soares por todos os ensinamentos que recebi e principalmente pela amizade e confiança em mim depositadas.
- a equipe do Laboratório TeleMídia da PUC-Rio pela amizade e ajuda indispensáveis durante a elaboração desta dissertação.
- a todos os meus colegas, agradeço a confiança e a amizade, sem dúvida fundamentais.
- a todos os professores e funcionários do Departamento de Informática da PUC-Rio que colaboraram para a conclusão deste trabalho.
- a CAPES, CNPq e Embratel pelo suporte financeiro fornecido durante o mestrado.

## RESUMO

A sincronização é uma tarefa importante na apresentação de documentos hipermídia, por esta razão, os sistemas hipermídia necessitam de ferramentas que possibilitem a definição de relacionamentos de sincronização temporal e espacial. Esta dissertação apresenta um Editor e Browser gráfico para Sincronização temporal e espacial de objetos multimídia/hipermídia (EBS) em documentos com composições aninhadas. O EBS, parte integrante do ambiente *HySEE* (HyperProp Show Editor and Executor), utiliza o Modelo de Contextos Aninhados (MCA) como modelo de estruturação e apresentação de dados, em conformidade com a proposta de padrão MHEG. O sistema permite a definição em forma gráfica da disposição temporal e espacial de um objeto, em relação a um tempo fixo ou a outros objetos, bem como a visualização das composições no tempo e espaço.

## ABSTRACT

Synchronization is a important task in the presentation of hypermedia documents, for that reason hypermedia systems need tools for the definition of temporal and spatial synchronization relationships. This thesis presents a graphic Editor and Browser (EBS) for temporal and spatial Synchronization of multimedia/hypermedia objects in documents with nested compositions. EBS, part of the *HySEE* environment (HyperProp Show Editor and Executor), uses the Nested Context Model (NCM) as the structural and presentation model, in conformance with the MHEG standard proposal. The system provides a graphical interface that permits the definition of the temporal and spatial placement of objects, relative either to a time axis or to other objects. It also permits the visualization of the compositions in time and space.

# Conteúdo

**Lista de Figuras** \_\_\_\_\_ **VI**

**Lista de Tabelas** \_\_\_\_\_ **IX**

**1. Introdução** \_\_\_\_\_ **1**

    1.1 Motivação \_\_\_\_\_ 1

    1.2 Objetivos \_\_\_\_\_ 3

    1.3 Organização da Dissertação \_\_\_\_\_ 4

**2. Trabalhos Relacionados** \_\_\_\_\_ **6**

    2.1 Introdução \_\_\_\_\_ 6

    2.2 Paradigmas \_\_\_\_\_ 6

    2.3 CMIFed \_\_\_\_\_ 9

        2.3.1 O Modelo de Dados do CMIF \_\_\_\_\_ 9

        2.3.2 Interface com o usuário \_\_\_\_\_ 11

    2.4 Firefly \_\_\_\_\_ 14

        2.4.1 O Modelo de Dados do Firefly \_\_\_\_\_ 15

        2.4.2 Interface com o Usuário \_\_\_\_\_ 17

2.5 Videobook	19
2.5.1 O Modelo de Dados do Videobook	20
2.5.2 Interface com o Usuário	22
2.6 Toolbook	22
2.6.1 Interface com o Usuário	23
2.7 Director	25
2.8 MAEstro	25
2.8.1 Interface com o Usuário	26
<b>3. O Sistema HyperProp</b>	<b>28</b>
3.1 Introdução	28
3.2 Arquitetura do HyperProp	29
3.3 O Modelo de Contextos Aninhados	31
3.4 O Subsistema de Apresentação do HyperProp	33
3.4.1 Eventos e Ações	34
3.4.2 Elos	36
3.4.3 Descritores	40
3.4.4 Objetos de Representação	41
3.4.5 Operações de Sincronismo	43
3.4.6 Browser de Base Privada	51
<b>4. O Editor e Browser de Sincronismo</b>	<b>53</b>
4.1 Introdução	53
4.2 A Interface Gráfica do EBS	55
4.2.1 A Time View	55

4.2.2 A Spatial View	67
4.3 Ambiente Integrado de Edição do HyperProp	73
<b>5. Implementação</b>	<b>76</b>
5.1 Requisitos do Sistema	78
5.2 Organização do Programa	78
5.3 Testes	83
5.4 A Interface do EBS no Ambiente UNIX/Motif	83
<b>6. Conclusões</b>	<b>85</b>
6.1 Comparaçao com Trabalhos Relacionados	86
6.2 Principais Contribuições	88
6.3 Trabalhos Futuros	89
<b>Referências Bibliográficas</b>	<b>91</b>

# **Lista de Figuras**

Figura 2.1: Interface genérica dos sistemas que utilizam timeline	7
Figura 2.2: A <i>hierarchy view</i> do CMIFed	12
Figura 2.3: A <i>channel view</i> do CMIFed	13
Figura 2.4: O <i>player</i> do CMIFed	14
Figura 2.5: Visualizador de item de mídia do sistema Firefly	17
Figura 2.6: Editor interativo de documentos do sistema Firefly	18
Figura 2.7: O espaço tridimensional utilizado no sistema Videobook	20
Figura 2.8: Correspondência entre um objeto cena e um gráfico temporal	22
Figura 2.9: A interface gráfica do sistema Toolbook	24
Figura 2.10: O Editor de Timeline do sistema MAEstro	27
Figura 3.1: Estrutura em camadas do sistema HyperProp	30
Figura 3.2: Arquitetura cliente/servidor do HyperProp	31
Figura 3.3: A Hierarquia de Classes do MCA	31
Figura 3.4: Nô D com perspectivas diferentes	32
Figura 3.5: Relação entre as regiões e eventos do nó A	35
Figura 3.6: Exemplo de um elo com relação (1:1)	37
Figura 3.7: Relação entre os eventos e as ações de um elo (n:m)	38
Figura 3.8: Exemplo da estrutura e definição de um elo (2:2)	39
Figura 3.9: Exemplos da especificação de descritores	41
Figura 3.10: Planos de armazenamento, de dados e de representação	42
Figura 3.11: Relações 1 e 2 de Allen modeladas com um elo de sincronização	44
Figura 3.12: Relações 3 e 4 de Allen modeladas com um elo de sincronização	45
Figura 3.13: Relações 5 e 6 de Allen modeladas com um elo de sincronização	45

Figura 3.14: Relações 7 e 8 de Allen modeladas com elos de sincronização	46
Figura 3.15: Relações 9 e 10 de Allen modeladas com um elo de sincronização	47
Figura 3.16: Relações 11 e 12 de Allen modelada com um elo de sincronização	47
Figura 3.17: Relação 13 de Allen modelada com elos de sincronização	48
Figura 3.18: Relações de alto nível	49
Figura 3.19: O browser de base privada do HyperProp	51
Figura 4.1: Uma visão geral do ambiente <i>HySEE</i> e do EBS	54
Figura 4.2: Layout da janela Time View do EBS	56
Figura 4.3: O objeto base da Time View	57
Figura 4.4: Um exemplo de uma seqüência temporal de nós	59
Figura 4.5: Inclusão automática de nós na Time View	59
Figura 4.6: Inclusão explícita de um nó na Time View	60
Figura 4.7: Inclusão de um nó na região de limbo	61
Figura 4.8: Criação de um elo de sincronização	61
Figura 4.9: Criação da relação <i>exibir ao término de</i>	62
Figura 4.10: Criação da relação exibir iniciando ao mesmo tempo	63
Figura 4.11: Criação da relação exibir iniciando e terminando ao mesmo tempo	64
Figura 4.12: Alteração na duração do nó O1	65
Figura 4.13: Repositionamento temporal do nó O2	66
Figura 4.14: A criação do elo <i>elo3</i> implica na destruição do elo <i>elo1</i>	67
Figura 4.15: A remoção do elo <i>elo1</i>	67
Figura 4.16: Layout da janela <i>Canal de Display</i>	69
Figura 4.17: Layout da janela <i>Canal de Áudio</i>	69
Figura 4.18: Relacionamento entre a Time View e a Spatial View	70
Figura 4.19: Repositionamento espacial de objetos na Spatial View	71
Figura 4.20: Redimensionamento espacial de objetos na Spatial View	72
Figura 4.21: Alteração do volume de exibição de objetos de áudio na Spatial View	72

Figura 4.22: O ambiente de manipulação de documentos hipermídia	73
Figura 4.23: Visão do usuário no ambiente integrado de edição e browsing	75
Figura 5.1: Estrutura modular do EBS	76
Figura 5.2: Esquema do ambiente integrado de edição e browsing do HyperProp	77
Figura 5.3: A hierarquia de classes do EBS	79
Figura 5.4: Notação para descrição das classes	80
Figura 5.5: A classe SeeWindow	80
Figura 5.6: A classe Presentation	81
Figura 5.7: A classe Channel	81
Figura 5.8: A classe Layer	82
Figura 5.9: A classe Object	82
Figura 5.10: A janela Time View no ambiente UNIX/Motif	84
Figura 5.11: A janela canal de display na Spatial View no ambiente UNIX/Motif	84
Figura 5.12: A janela canal de áudio na Spatial View no ambiente UNIX/Motif	84

# **Lista de Tabelas**

Tabela 3.1: Ações que podem ser atribuídas a uma região de um nó	36
Tabela 3.2: Operadores de um ponto de encontro	38
Tabela 3.3: Relações 1 e 2 da álgebra de intervalos de Allen	43
Tabela 3.4: Relações 3 e 4 da álgebra de intervalos de Allen	44
Tabela 3.5: Relações 5 e 6 da álgebra de intervalos de Allen	45
Tabela 3.6: Relações 7 e 8 da álgebra de intervalos de Allen	45
Tabela 3.7: Relações 9 e 10 da álgebra de intervalos de Allen	46
Tabela 3.8: Relações 11 e 12 da álgebra de intervalos de Allen	47
Tabela 3.9: Relação 13 da álgebra de intervalos de Allen	47
Tabela 6.1: Comparação entre alguns sistemas de edição de sincronização	87

# Capítulo 1

## Introdução

### 1.1 Motivação

Documentos multimídia/hipermídia são documentos não-convencionais compostos por várias mídias, como áudio, vídeo, imagens gráficas, textos, etc. Esses documentos possuem grande significado no que diz respeito a comunicação, pois contêm informação codificada na forma de diferentes mídias, as quais podem estar interrelacionadas dentro do documento, diferentemente do antigo paradigma linear dos documentos convencionais utilizados. Nesta dissertação iremos tratar de documentos interativos, isto é, documentos que possuem tanto comportamento de apresentação previsível (síncrona), quanto não previsível (assíncrona, pois podem depender da interação do usuário).

O termo *apresentação* será utilizado nesta dissertação para denotar a exibição das várias mídias ao usuário. A apresentação das mídias podem ser classificadas conforme seu comportamento de exibição em: apresentação dinâmica (mídias áudio e vídeo); e apresentação estática (mídias texto e imagens gráficas). Para as mídias gráficas e textuais (imagens gráficas e texto), apresentação implica em exibição no monitor de vídeo. No caso das mídias audiovisuais (áudio e vídeo), apresentação indica exibição auditiva e visual da informação via dispositivos de E/S apropriados.

A apresentação de um documento multimídia/hipermídia, normalmente, é constituída por um conjunto de componentes dinâmicos e estáticos: imagens, vídeos, textos, trechos de áudio, incluindo outras apresentações já editadas.

Sistemas de autoria multimídia/hipermídia podem ser definidos como sistemas que permitem a criação, o armazenamento e a recuperação de documentos dessa natureza. A tarefa de criação desses documentos não é uma tarefa fácil, pois é necessário fornecer ao autor uma completa manipulação de sua estrutura, permitindo que os relacionamentos entre os diversos componentes sejam definidos de maneira amigável. Como, em geral, a estrutura de um documento multimídia/hipermídia não é simples, para visualizá-la necessita-se de ferramentas gráficas [Much96] [MuSC95] [MuSo95], que exibam de maneira simplificada suas diversas conexões, com o objetivo de reduzir o problema clássico denominado “perdido no hiperespaço”.

Entre os relacionamentos de um documento multimídia/hipermídia se destacam as relações de sincronização. *Sincronização* se refere aos mecanismos usados na ordenação de eventos no domínio do tempo e do espaço, onde um *evento* pode ser definido, simplificadamente, pela exibição ou seleção de um trecho (região) do documento. Para que um sistema hipermídia possa definir a sincronização de uma apresentação, ele deve possuir uma representação interna adequada, e fornecer um mecanismo que capture os relacionamentos de sincronização entre os componentes.

Suponha como exemplo, um documento multimídia que corresponda a uma aula. Durante a explanação, descrições e ilustrações devem ser apresentadas ao usuário (aluno) para permitir um melhor entendimento do assunto. É necessário, portanto, definir relacionamentos temporais entre a voz (áudio) do professor e as ilustrações (imagens gráficas, textos e possivelmente vídeos) que são referenciadas, para manter a consistência temporal da apresentação do documento. Ao alinhamento desses relacionamentos no domínio do tempo dá-se o nome de *sincronização temporal*. Sistemas hipermídia devem fornecer mecanismos para definir tais relacionamentos.

A definição de relações de sincronismo temporal dentro de um documento aumenta consideravelmente seu poder de comunicação. É importante notar que essas interconexões devem ser definidas relacionando os componentes do documento entre

si, e não em relação ao próprio eixo do tempo. Isso se deve ao fato que, durante a apresentação do documento, podem ocorrer pontos onde poderão ser capturadas ações dos espectadores, determinando como as apresentações devem prosseguir, onde não se pode prever o instante do seus acontecimentos. Outro motivo pelo qual o relacionamento temporal entre os componentes de um documento deve ser relativo, são os reflexos de mudanças na velocidade da exibição quando uma apresentação é executada em máquinas diferentes. Nesse caso, torna-se difícil garantir que os componentes serão exibidos nos instantes definidos por um tempo absoluto. Outros fatores, apresentados no decorrer do texto, reforçarão ainda mais as vantagens do sincronismo relativo entre os componentes.

A *sincronização espacial* é definida pela utilização dos dispositivos de E/S de uma plataforma de exibição, em um dado instante de tempo, para a apresentação de um documento multimídia/hipermídia. No caso do dispositivo de display (monitor de vídeo), a sincronização espacial irá definir como os componentes de um documento estarão dispostos (posição ocupada na tela, prioridade de sobreposição, etc.). Em se tratando do dispositivo de áudio, a sincronização espacial define, por exemplo, qual o percentual de volume utilizado por cada componente do documento em uma mixagem.

## 1.2 Objetivos

O principal objetivo desta dissertação é implementar um sistema editor e browser gráfico para definição da sincronização temporal e espacial de documentos multimídia/hipermídia em conformidade com o padrão MHEG. O sistema desenvolvido, chamado de EBS, se constitui em uma das ferramentas do sistema HyperProp [Soar95] utilizado como base.

O sistema HyperProp combina características de modelagem conceitual de dados hipermídia, descritos através do Modelo de Contextos Aninhados [SCR95], com a utilização de uma arquitetura aberta que distingue os componentes de exibição dos componentes de dados. A utilização combinada desses componentes facilita os processos de edição e apresentação dos documentos multimídia/hipermídia.

O sistema HyperProp possui ferramentas gráficas que visam auxiliar a navegação e legibilidade da estrutura dos documentos. As ferramentas de browsers e trilhas propostas em [Much96] tentam minimizar o problema da desorientação do usuário durante o processo de manipulação de documentos, explorando mecanismos de navegação através de uma apresentação gráfica da estrutura.

A necessidade de outra ferramenta gráfica onde o usuário possa definir as relações temporais e espaciais entre os componentes de um documento multimídia/hipermídia de uma maneira natural deu origem ao EBS. A construção da ferramenta baseou-se no estudo de algoritmos que possibilitam a definição e manutenção da consistência da sincronização durante o processo de edição. O sistema fornece um mecanismo de compilação incremental, validando a sincronização especificada pelo autor, passo a passo.

A definição de operações explícitas de sincronismo entre os componentes de um documento, tais como: *exiba em paralelo*, *exiba ao término de*, são exemplos de alguns algoritmos estudados e enfocados nessa dissertação.

O EBS juntamente com o browser gráfico de estrutura apresentado em [Much96] fornecem um ambiente completo para a criação e manipulação de documentos hipermídia através de suas visões estrutural, temporal e espacial.

### **1.3 Organização da Dissertação**

Esta dissertação está organizada da seguinte forma.

No segundo capítulo são apresentados os principais paradigmas de edição e apresentação da sincronização temporal e espacial de documentos multimídia/hipermídia. Alguns sistemas existentes que utilizam esses paradigmas são discutidos. Com base na análise dos diversos paradigmas, identificam-se as características importantes abordadas pelos sistemas analisados, que serviram de base para a proposta do sistema de edição de sincronismo temporal e espacial desta dissertação.

No terceiro capítulo é apresentado o sistema hipermídia *HyperProp*, para o qual foi desenvolvida a ferramenta gráfica. Nesse capítulo são apresentados a arquitetura do sistema HyperProp, o modelo conceitual básico utilizado, o *Modelo de Contextos Aninhados* (MCA), e o subsistema de apresentação do HyperProp, enfatizando as classes do modelo conceitual estendido que definem os aspectos referentes à sincronização temporal e espacial de um hiperdocumento. No capítulo, são introduzidos também os conceitos de operações básicas de sincronismo, e como elas são definidas no modelo conceitual estendido do sistema HyperProp.

O quarto capítulo é dedicado a apresentação do foco principal desta dissertação: o *Editor e Browser Gráfico de Sincronismo* (EBS), uma ferramenta gráfica para a edição e apresentação da sincronização temporal e espacial de um documento multimídia/hipermídia no sistema HyperProp. Seus requisitos e objetivos são apresentados baseados nas características analisadas dos diversos trabalhos relacionados. As visões temporal e espacial do usuário em relação a um hiperdocumento são mostradas e suas funções detalhadas nesse capítulo.

A estratégia da implementação utilizada no EBS é apresentada no quinto capítulo, onde são mostrados a hierarquia de classes do sistema, a estrutura modular do programa e os principais algoritmos utilizados para o seu desenvolvimento.

Finalmente, o sexto capítulo é reservado para a apresentação e crítica das conclusões analisadas, ressaltando as principais contribuições desta dissertação e propondo algumas sugestões para trabalhos futuros complementares. Nesse capítulo é realizada também uma comparação entre os sistemas estudados e o EBS, analisando-se as vantagens e desvantagens de cada um segundo critérios específicos.

# Capítulo 2

## Trabalhos Relacionados

### 2.1 Introdução

Este capítulo apresenta alguns sistemas de autoria hipermídia que possuem ambientes gráficos para definição da sincronização de documentos multimídia/hipermídia. Os sistemas apresentados foram escolhidos entre os vários existentes, para exemplificar os principais paradigmas de edição e definição de sincronismo.

Inicialmente, serão descritos os principais paradigmas a partir dos quais foram classificados os sistemas com respeito às funções de definição dos aspectos de sincronização.

### 2.2 Paradigmas

Os paradigmas mais importantes para especificação do sincronismo, conforme visto em [HRB93b] [BuZ93b], são:

- *timeline*;
- *scripting*;
- baseado em eventos (pontos de referência).

O paradigma *timeline* possui como principal característica o posicionamento dos componentes de um documento multimídia/hipermídia em relação a um eixo do tempo. Nesse paradigma todos os relacionamentos de sincronização não possuem vinculação relativa a outros componentes, mas estão vinculados de forma absoluta ao próprio eixo do tempo. Nessa estratégia, o autor especifica manualmente o início e a duração de cada um dos componentes da apresentação. A Figura 2.1 mostra como é a interface, de uma maneira geral, dos sistemas baseados no paradigma de timeline.

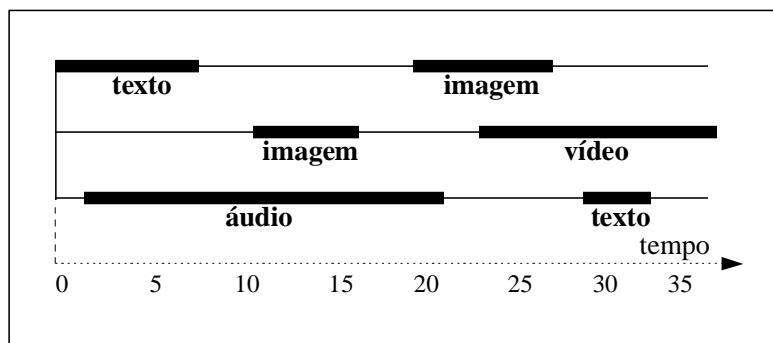


Figura 2.1: Interface genérica dos sistemas que utilizam timeline

O propósito desse paradigma utiliza uma técnica bastante intuitiva, que é também sua principal vantagem, onde o próprio decorrer do tempo é responsável pela ativação das mudanças de comportamento de apresentação e pelo acionamento dos relacionamentos temporais definidos no documento.

A estratégia timeline possui várias desvantagens, dentre elas, as principais são:

- uma simples alteração num segmento da apresentação pode requerer que todos os tempos definidos para eventos posteriores tenham que ser atualizados manualmente;
- é impossível nesse paradigma o posicionamento relativo de objetos de duração indefinida
- é impossível a representação de relacionamentos que dependem de interações com o usuário.

Como exemplo de um sistema que utiliza essa abordagem, serão apresentados na Seção 2.8 aspectos gerais da implementação do sistema hiperídia *MAEstro*. Outros

exemplos de sistemas que também utilizam dessa abordagem são: *MacroMind Action*, *Animation Works Interactive* etc.

O segundo paradigma leva em consideração que a tarefa de construção da apresentação de um documento multimídia/hipermídia é realizada através de uma linguagem de especificação do seu comportamento, chamada de *scriptware*.

A linguagem *scriptware* é parecida com as linguagens de programação convencionais, onde todos os relacionamentos de sincronização são explicitados de forma textual. Apresentações curtas podem ser facilmente construídas com esta estratégia porém, a programação e manutenção de apresentações de grande porte tornam-se tarefas complexas, que é considerada a principal desvantagem dessa técnica. Como exemplo de um sistema que utiliza essa abordagem, serão apresentados na seção 2.6, aspectos gerais da implementação do sistemas hipermídia *Asymetrix Toolkit*. Outros exemplos de sistemas que também utilizam dessa abordagem são: *Metacar*, *HyperStudio*, *Linkway Live*, etc.

Na Seção 2.5 será apresentado o sistema *Videobook* que utiliza a idéia de scripts visuais. Nesse sistema, o usuário não programa textualmente os scripts, mas utiliza uma interface gráfica para fazê-lo.

O EBS está inserido no grupo de sistemas que usam o paradigma *baseado em eventos*. Um *evento* pode ser definido como a ocorrência (exibição ou seleção) de uma região do documento previamente especificada pelo autor.

Segundo este paradigma, o autor seleciona trechos nos documentos cuja exibição ou seleção caracteriza a ocorrência de um evento. Em seguida, o sincronismo da apresentação é realizado através do relacionamento desses eventos. O EBS utiliza entidades denominadas de *elos* para definir a interconexão (relacionamento) entre os componentes de um documento. As entidades utilizadas para definir o relacionamento, dentro de um documento, em outros sistemas, em geral, são chamadas de nomes diferentes. O sistema *CMIFed* [RoJMB93], que será apresentado na Seção 2.3, denomina essas entidades de *syncarcs* e *hyperlinks*. Já o sistema *Firefly* [BuZe92], mostrado na Seção 2.4, os chama de *temporal constraints*.

## 2.3 CMIFed

O CMIFed (Editor CMIF) é um ambiente de edição e apresentação para documentos hipermídia. Ele foi desenvolvido pelo núcleo de sistemas multimídia do CWI, Centrum voor Wiskunde en Informatica, da Holanda. A sigla CMIF significa “CWI Multimedia Interchange Format”. O termo é utilizado para representar os dois principais aspectos abordados por esse ambiente: a estrutura e a apresentação. O modelo conceitual de dados utilizado pelo CMIFed, o *CMIF hypermedia model*, é um modelo hipermídia baseado em outro modelo conceitual, o *Amsterdam hypermedia model* [HRB93a], que é uma extensão do modelo de referência Dexter [HaSc90].

O modelo CMIF permite que uma apresentação seja recursivamente construída por uma série de subapresentações. Uma apresentação no CMIF pode ser denominada:

- *apresentação composta*, quando ela contém outras apresentações aninhadas;
- *apresentação atômica*, quando ela não contém nenhuma apresentação na sua subárvore de aninhamento.

Um documento é, portanto, uma árvore com hierarquia de apresentações compostas, onde as folhas dessa árvore são apresentações atômicas [HRB93b] [RJMB93].

### 2.3.1 O Modelo de Dados do CMIF

Um *evento*, no modelo conceitual CMIF, é definido como o menor fragmento de mídia que pode ser manipulado pelo sistema. São exemplos de eventos: pequenos fragmentos de vídeo, áudio, imagem ou texto. Uma apresentação atômica é uma coleção de eventos. *Marcadores* são utilizados como pontos de ancoragem das conexões entre componentes do documento. Esses marcadores ficam armazenados juntamente com os dados.

Os eventos que compõem uma apresentação atômica estão relacionados através de relacionamentos temporais (“*timing constraints*”). São usados dois mecanismos para definir os relacionamentos temporais:

- composição paralela e seqüencial;
- e arcos de sincronização, ou simplesmente sinc-arcs.

A própria construção da árvore de hierarquia de uma apresentação define os relacionamentos temporais através dos mecanismos de composições paralela e seqüencial. O posicionamento dos nós dentro da árvore de apresentação possui relações de tempo, e sua estrutura define o comportamento temporal de sua exibição.

Além da definição de relacionamentos temporais a partir da estrutura hierárquica dos componentes de uma apresentação, através do modelo CMIF, pode-se definir relacionamentos entre quaisquer nós de uma mesma sub-árvore através dos *sinc-arcs* (arcos de sincronização). Um *sinc-arc* é uma relação entre marcadores em dois eventos de uma mesma apresentação atômica, que especifica um determinado valor de retardo. Suponha, como exemplo de sua utilização, que em uma sub-árvore de uma apresentação atômica foi especificado que um nó de vídeo deve ser apresentado em paralelo com uma música (através do posicionamento paralelo dos dois nós na árvore), e que nós desejamos retardar a exibição do início da música de 2 segundos. Isto pode ser feito pela adição de um sinc-arc do evento inicial do nó de vídeo para o evento inicial do nó de áudio (música), atribuindo a ele um retardo de 2 segundos.

O modelo CMIF também define os *elos de hipermídia* (*hyperlinks*), que são relacionamentos entre fragmentos de informações que dependem de um evento de interação do usuário. Âncoras são definidas como uma parte de um item de dados. Os hyperlinks são conexões entre duas âncoras, origem e destino do elo.

Outro conceito importante do CMIF é o conceito de canais. Um *canal* é uma abstração de características específicas de um dispositivo de saída da plataforma de exibição, e é constituído por um conjunto de eventos. Cada canal possui um tipo de mídia específico que deve ser do mesmo tipo dos eventos que estão situados nele. Um canal de áudio define, por exemplo, qual o volume de exibição dos componentes de dados que estão contidos nele. Um canal de imagens gráficas pode especificar qual a posição na tela ocupada pelos seus componentes, etc.

Todos os elementos do modelo CMIF (eventos, canais, nós de composição, sinc-arcs, âncoras e elos) possuem uma lista de atributos que descrevem suas propriedades, como: nome, duração de exibição, etc. Essas propriedades estão vinculadas ao tipo de mídia e à natureza de cada elemento.

O objetivo do projeto do CMIFed foi desenvolver uma solução para sistemas hipermídia que utilizam a noção de documentos estruturados. O ambiente foi desenvolvido para suportar a composição hierárquica de apresentações. O processo de criação de apresentações compostas pode ser dividido em dois estágios separados:

- a construção da estrutura de uma apresentação;
- a definição de relacionamentos temporais entre os componentes de dados.

Esses processos serão melhor explicados na próxima seção quando forem apresentadas as interfaces com o usuário.

### 2.3.2 Interface com o usuário

O ambiente de autoria hipermídia do CMIFed descreve uma interface com o usuário que possui como objetivo fornecer três diferentes visões de um mesmo documento multimídia. Utilizando-se dessas visões, o autor pode manipular todos estágios da tarefa de criação da apresentação de um documento em um único ambiente de interface integrado. As visões do ambiente CMIFed são [HRB93b] [RJMB93]:

- a *hierarchy view*;
- a *channel view*;
- e o *player*.

A *hierarchy view* oferece a edição e visualização da estrutura da apresentação. Ela é a primeira janela de autoria que o autor visualiza, onde ele pode construir a estrutura de um documento multimídia utilizando os paradigmas *top-down* ou *bottom-up*. Na *hierarchy view*, são especificadas as relações temporais de composição paralela e composição seqüencial, descritas anteriormente.

O autor compõe seu documento montando a estrutura hierárquica definida pelo aninhamento das subapresentações, podendo navegar dentro da estrutura hierárquica do documento e “entrar” dentro das composições. A Figura 2.2 mostra a janela de interface que fornece a visão hierárquica (hierarchy view) de um documento no CMIFed [HRB93b] [RJMB93].

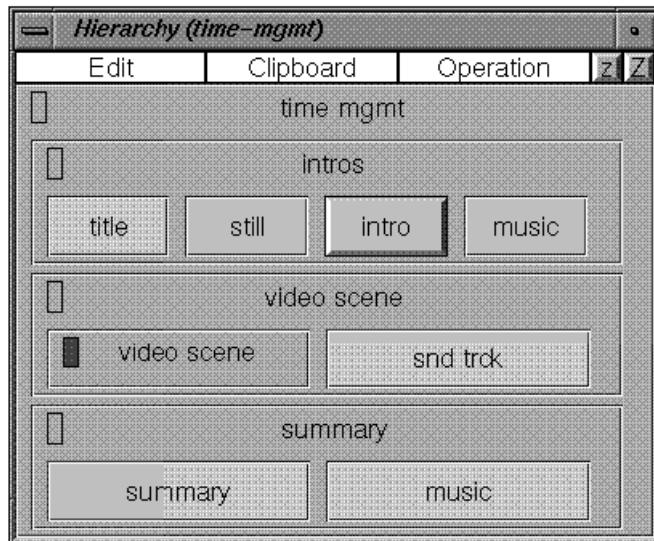


Figura 2.2: A *hierarchy view* do CMIFed

É possível ver na Figura 2.2 que a apresentação “time-mgmt”, mostrada como exemplo, é constituída de várias subapresentações. A árvore de hierarquia das apresentações compostas envolvidas nesse exemplo possui como nó raiz, a apresentação composta chamada “time-mgmt”, que é o retângulo mais externo que contém todas as demais subapresentações. A apresentação “time-mgmt”, por sua vez, é constituída pelas apresentações “intros”, “video scene”, “summary”, que também são apresentações compostas. Note que a ordem em que as subapresentações estão dispostas na janela (de cima para baixo) define a sincronização temporal por composição seqüencial do modelo CMIF. Ou seja, a apresentação “intros” será apresentada antes da apresentação “video scene”, que por sua vez será exibida antes da apresentação “summary”.

Ainda nesse exemplo, a apresentação “video scene” possui dois componentes: a apresentação composta “video scene” e um nó de música chamado “snd trck”. Esses dois componentes estão dispostos um ao lado do outro dentro da janela da

apresentação composta “video scene” principal, que simboliza que esse dois componentes serão exibidos em paralelo no tempo (ao mesmo tempo). Essa definição de paralelismo é um exemplo da sincronização através de composições paralelas do modelo CMIF.

A *channel view* fornece uma visão baseada nos canais criados pelo autor durante o processo de definição de uma apresentação multimídia/hipermídia. A Figura 2.3 mostra como exemplo, a janela do CMIFed que fornece a visão de canais (channel view) da apresentação “time-mgmt”.

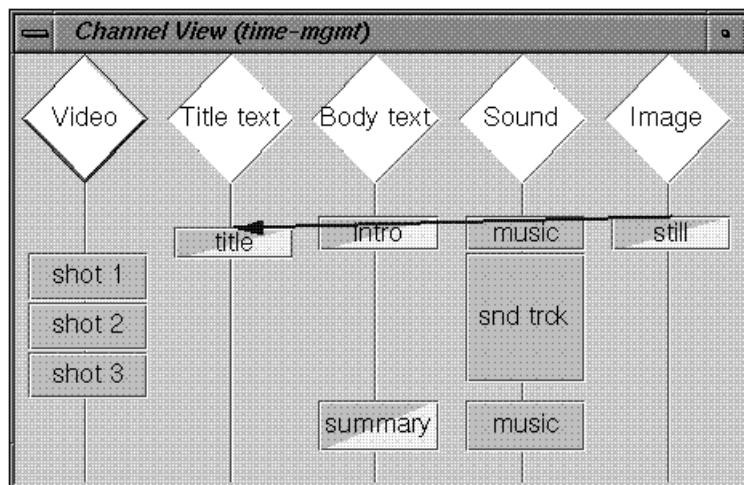


Figura 2.3: A *channel view* do CMIFed

No channel view da Figura 2.3, são mostrados os canais utilizados na apresentação “time-mgmt”, que são: “Video”, “Title text”, “Body text”, “Sound” e “Image”. O posicionamento dos eventos dentro dos canais estão relacionados aos momentos de suas ocorrências. Os eventos estão alinhados seqüencialmente no tempo dentro dessa visão. A ordem de suas disposições dentro dos canais no sentido vertical define o alinhamento temporal (de cima para baixo). Ou seja, o evento “shot 1” ocorre antes do evento “shot 2”, que ocorre antes do evento “shot 3”.

Os relacionamentos temporais definidos na channel view são definidos através dos sinc-arcs, que podem ser criados e alterados dentro dessa visão. No exemplo da Figura 2.3, foi criado um sinc-arc do evento “still” do canal “Image”, para o evento “title” do canal “Title text”. Esse sinc-arc é representado por uma seta que conecta os dois eventos (Figura 2.3).

A terceira visão da apresentação, fornecida pelo *player* (exibidor), mostra o efeito da exibição de uma determinada apresentação em uma plataforma específica. O player permite que o autor possa editar os aspectos de layout da apresentação, como a geometria das janelas usadas por canais que utilizam o dispositivo de vídeo. O autor também pode definir e posicionar os pontos de ancoragem em imagens estáticas da apresentação. Um exemplo de player é mostrado na Figura 2.4.

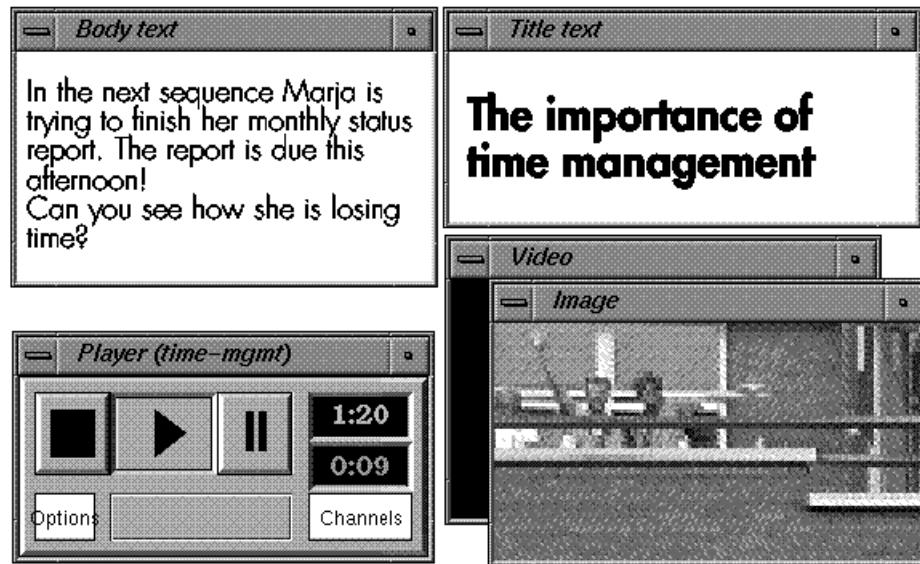


Figura 2.4: O *player* do CMIFed

O player possui uma janela de controle (positionada no canto inferior esquerdo). A função dessa janela é controlar o tempo de exibição que está sendo mostrado na sua visão. O layout das janelas pode ser modificado, e as alterações de comportamento dos nós são armazenadas dentro dos canais das mídias.

## 2.4 Firefly

O sistema multimídia/hipermídia Firefly foi desenvolvido pelo grupo de estudos em sistemas multimídia do Laboratório de Ciências de Informação do Centro de Pesquisas da Xerox em Palo Alto (Xerox PARC), Califórnia [BuZe92] [BuZ93a] [BuZ93b]. O objetivo do sistema Firefly foi fornecer facilidades para criação, edição, manutenção e apresentação de documentos multimídia/hipermídia. Um protótipo desse sistema foi implementado em estações de trabalho SUN no ambiente de programação Cedar.

O sistema Firefly consiste de três componentes:

- ferramentas de autoria para a criação e edição de relacionamentos temporais;
- um escalonador de tarefas em tempo de compilação para preprocessamento das especificações temporais;
- um escalonador de tarefas em tempo de execução para controlar a apresentação de documentos.

Nesta Seção serão analisadas as principais características das ferramentas de autoria para a criação e edição de relacionamentos temporais.

#### **2.4.1 O Modelo de Dados do Firefly**

No modelo de dados utilizado pelo sistema Firefly, um documento multimídia/hipermídia consiste de três partes principais [BuZe92]:

- itens de mídia;
- relacionamentos temporais de sincronização (*temporal synchronization constraints*);
- e listas de operações.

Um *item de mídia* corresponde a um fragmento de informação de um tipo de mídia específico. Um item de mídia pode ser: um arquivo de texto, um clip de vídeo, um arquivo de áudio, etc. Um item de mídia é inserido em um documento por referência, para permitir que um mesmo item de mídia seja incorporado em outros documentos, ou até mesmo, inserido repetidamente dentro de um mesmo documento. O próprio item de mídia pode descrever como será seu comportamento temporal de exibição. Um item de mídia possui quatro componentes:

- um *indicador do tipo de mídia*.
- *eventos*, que representam os pontos em que a apresentação do item de mídia pode ser sincronizada com outros itens de mídia.
- *procedimentos*, que operam sobre um item de mídia e seus eventos.

- um ponteiro, que é uma referência para a informação descrita pelo item de mídia.

Os eventos podem ser de dois tipos: *eventos síncronos*, quando seu posicionamento temporal pode ser previsto a priori; ou *eventos assíncronos*, quando não se pode prever com precisão sua ocorrência no tempo.

A principal categoria de procedimentos é chamada de *procedimentos de controle*. Eles atuam sobre o comportamento de exibição do item de mídia. Por exemplo, um item de mídia do tipo áudio pode possuir procedimentos de controle que permitam aumentar ou diminuir o volume de exibição do áudio em um determinado instante. Os eventos de início de apresentação, fim de apresentação e pausa, são exemplos de alguns eventos que, necessariamente, devem possuir procedimentos de controle definidos.

Os *relacionamentos temporais de sincronização* (“temporal synchronization constraints”) especificam como é realizada a sincronização entre pares de eventos em um ou mais itens de mídia. O sistema Firefly oferece duas classes de relacionamentos temporais de sincronização:

- *igualdades temporais* (“temporal equalities”);
- e *desigualdades temporais* (“temporal inequalities”).

Os relacionamentos de igualdades temporais podem definir que dois eventos ocorram simultaneamente no tempo, ou que dois eventos ocorram seqüencialmente no tempo com um intervalo de tempo fixo especificado entre os dois.

As relações de desigualdades temporais servem para relacionar seqüencialmente dois eventos no tempo, com um intervalo de tempo indeterminado entre eles.

*Listas de operações* podem ser associadas aos eventos de um item de mídia. Elas permitem controlar o comportamento de exibição do próprio item de mídia, ou de outros itens de mídia através de operações de troca de mensagens entre os itens de mídia durante a apresentação de um documento. As listas de operações, ao contrário dos procedimentos de controle, não são armazenadas juntamente com os itens de mídia, para permitir que sejam reutilizadas por outros itens de mídia.

## 2.4.2 Interface com o Usuário

O sistema Firefly oferece duas ferramentas para realizar as tarefas de criação, visualização e edição de especificações temporais. São elas [BuZe92] [BuZ93a] [BuZ93b]:

- um visualizador de itens de mídia;
- e um editor interativo de documentos.

A Figura 2.5 mostra dois visualizadores da mídia áudio. Na parte superior da figura aparece o editor de áudio TiogaVoice. A parte inferior mostra o visualizador temporal de itens de mídia do sistema Firefly.

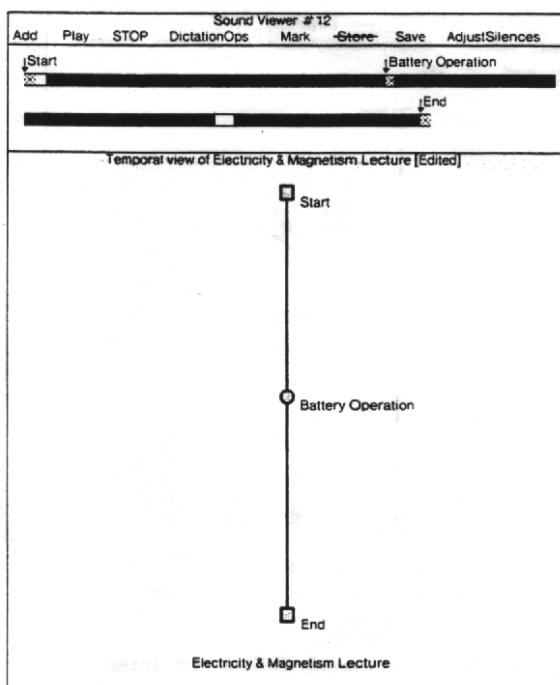


Figura 2.5: Visualizador de item de mídia do sistema Firefly

No exemplo da Figura 2.5 o visualizador de mídia mostra um exemplo da visualização de um item da mídia áudio chamado “Electricity & Magnetism Lecture”. Os eventos de início e de fim de exibição são representados por nós retangulares que possuem os nomes “Start” e “End”. O evento interno “Battery Operation” é representado por um nó circular posicionado entre os eventos de início e fim do item de mídia.

Tanto o visualizador de item da mídia, quanto o editor interativo de documentos fornecem uma visualização que utiliza uma notação em grafos específica. Na notação em grafos do Firefly, os nós retangulares representam os eventos de início e de fim de um item de mídia, enquanto os nós circulares representam eventos internos. As linhas representam o tempo decorrido entre dois eventos, onde seu comprimento é proporcional à duração entre os eventos.

A interface gráfica da outra ferramenta do sistema Firefly, o editor interativo de documentos, é mostrada na Figura 2.6.

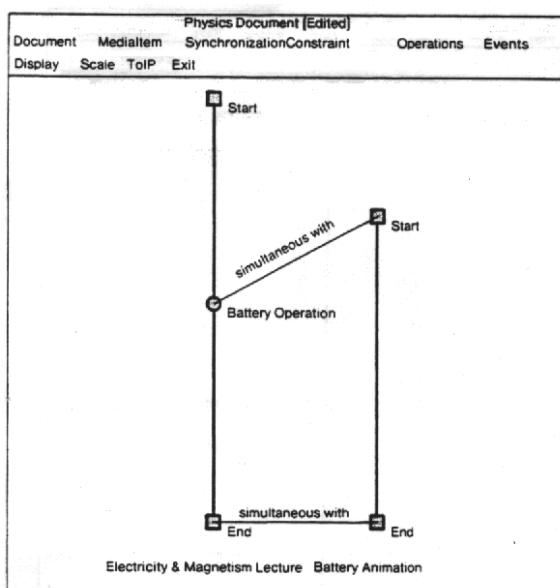


Figura 2.6: Editor interativo de documentos do sistema Firefly

O editor, ilustrado na Figura 2.6, possui duas regiões de interface. A região de menu, que contém funções para criação e edição de documentos, e a região de edição gráfica, que contém uma visão temporal do documento na notação de grafos.

No exemplo da Figura 2.6 é mostrado um documento que contém dois itens de mídia: um item de áudio intitulado “Electricity & Magnetism Lecture”, e um item de vídeo intitulado “Battery Animation”, que contém uma animação. Para exibir a animação durante a exibição do áudio, foi necessário adicionar dois relacionamentos temporais de sincronização. O primeiro relacionamento, de igualdade temporal, conecta o evento “Battery Operation” do áudio com o evento de início (“Start”) da animação, especificando que eles devem ocorrer ao mesmo tempo. O segundo relacionamento,

também de igualdade temporal, conecta o evento de fim do áudio (“End”) com o evento de fim (“End”) da animação, especificando que eles devem ocorrer ao mesmo tempo, ou seja, o áudio e a animação devem terminar juntos.

Na interface do editor, um item de mídia pode ser inserido em um documento ativando a opção de menu e determinando o item de mídia a ser inserido. Eventos assíncronos também podem ser definidos dentro dos itens de mídia, através da interface com o editor do sistema Firefly.

## 2.5 Videobook

Esta seção apresenta o sistema hipermídia *Videobook* que utiliza uma modelagem de dados baseada em cenários. Esse sistema é fruto das investigações do grupo de estudos em hipermídia do Laboratório de Pesquisas C&C, da NEC Corporation do Japão. O principal objetivo da implementação desse sistema foi construir uma ferramenta gráfica que permita a criação e edição de documentos multimídia/hipermídia baseados no modelo [OgHK90].

O modelo de dados hipermídia utilizado traz a idéia de composições de nós com scripts visuais. As principais características desse modelo são:

- Um nó de composição é composto por nós que contêm informação e botões, estruturados de acordo com um script que especifica seus parâmetros de apresentação, como, por exemplo, o layout da janela de exibição e o tempo de exibição.
- Os parâmetros são expressos visualmente através de objetos de três dimensões, e o script de apresentação vai sendo construído, e pode ser visualizado, observando-se diretamente o layout gráfico dos objetos.
- Os botões são objetos especiais que servem para sincronizar os dados e também servem para auxiliar a navegação dentro do documento.

A edição e visualização dos parâmetros de apresentação em um plano tridimensional baseiam-se nos seguintes fatos:

- Para apresentar diferentes dados de maneira síncrona, necessita-se especificar quando (relação temporal) e onde (relação espacial) exibi-los.
- É necessário utilizar uma nova metáfora para compreender como uma seqüência de dados baseada no tempo pode ser representada por um objeto visual.

O modelo utilizado pelo sistema Videobook possui como objetivo a criação de uma representação visual de um objeto baseada em cenários (scripts com características áudio-visuais), e construir um mecanismo de criação de relacionamentos temporais sobre esta nova representação.

### 2.5.1 O Modelo de Dados do Videobook

Para descrever o modelo utilizado pelo sistema Videobook, inicialmente é necessário definir o layout do espaço tridimensional utilizado como referência, conforme ilustrado na Figura 2.7 [OgHK90].

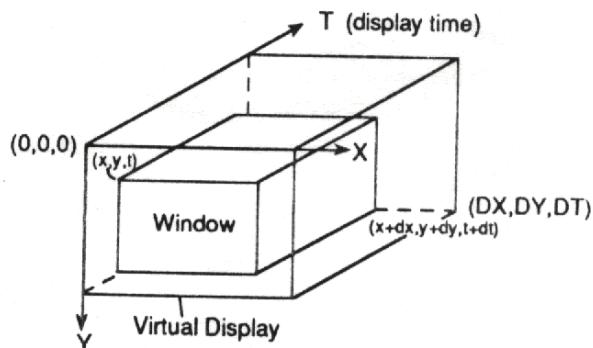


Figura 2.7: O espaço tridimensional utilizado no sistema Videobook

Na Figura 2.7, o plano (X,Y) corresponde a tela virtual do monitor de vídeo, enquanto que o eixo T corresponde ao eixo do tempo. A coordenada T especifica o início e o fim das exibições dos componentes da apresentação. A origem do eixo do tempo ( $T=0$ ) indica o início da apresentação completa. No espaço, os componentes da apresentação são representados por objetos sólidos.

As classes de objetos componentes do modelo de dados utilizado no sistema Videobook são as seguintes:

- mídia;

- disparador (“trigger”);
- e cena.

Um objeto *mídia* contém um fragmento de informação que pode ser classificados como áudio, vídeo, texto, imagem gráfica, etc. Um objeto mídia possui os parâmetros (dx, dy, dt) que indicam a dimensão da janela na tela do monitor de vídeo (dx, dy), e a duração de exibição (dt).

Um objeto da classe *disparador* serve para relacionar objetos do tipo mídia. No objeto disparador, um botão (origem do relacionamento) envia uma mensagem causando uma ação determinada sobre um objeto mídia específico (destino do relacionamento).

Os botões de um objeto da classe disparador também são representados por objetos sólidos, e seus parâmetros (dx, dy, dt) indicam a região de ancoragem do botão (dx, dy) e a duração da exibição do botão (dt).

Um objeto *cena* corresponde a um nó de composição que contém a apresentação de uma coleção de objetos mídia, objetos disparador, e de outros objetos do tipo cena. Uma cena possui um script que é construído visualmente através do posicionamento de seus componentes dentro do espaço tridimensional. Os objetos do tipo cena podem estar aninhados através de uma estrutura hierárquica.

O script visual construído pela montagem de um objeto cena pode ser convertido em um script textual convencional, ou em um gráfico temporal que especifica os tempos de exibição dos componentes de uma apresentação. A Figura 2.8 ilustra a representação visual de uma cena e a representação correspondente do seu gráfico temporal [OgHK90].

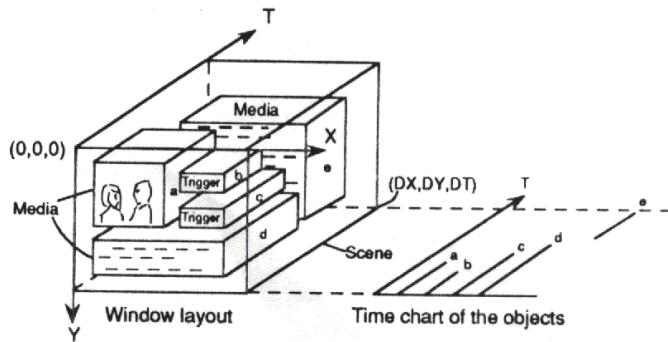


Figura 2.8: Correspondência entre um objeto cena e um gráfico temporal

### 2.5.2 Interface com o Usuário

No sistema Videobook existe uma ferramenta gráfica que oferece uma interface para permitir que a edição de objetos do tipo cena seja realizada de maneira visual. O editor de cenas, como é chamado, mostra graficamente o script visual, o gráfico temporal correspondente de uma determinada cena, e o layout da tela de exibição de um determinado momento da apresentação. Essas três visões tentam facilitar a montagem do script da apresentação de uma cena.

O editor de cenas foi construído para ser a principal interface de autoria de documentos do sistema Videobook. A sincronização de uma cena pode ser editada e modificada durante o processo de autoria tanto de forma gráfica, através do posicionamento dos objetos no espaço, como de maneira textual, editando manualmente o script textual que descreve o sincronismo da cena.

Para editar o conteúdo de um determinado objeto mídia específico, basta clicar sobre o objeto desejado e o editor daquele tipo de mídia será acionado.

## 2.6 Toolbook

O sistema *Toolbook* foi desenvolvido pela Asymetrix Corporation dos Estados Unidos. Trata-se de um sistema de autoria hipermídia similar em muitos aspectos ao seu antecessor, o HyperCard. Ambos servem para o desenvolvimento dos mesmos tipos de aplicações hipermídia.

O HyperCard utiliza uma metáfora baseada em cartões (*cards*); o Toolbook utiliza uma metáfora equivalente, porém baseada em páginas. Um documento multimídia/hipermídia no Toolbook é chamado de um *livro*. Um livro Toolbook é constituído por um conjunto de páginas que são construídas, passo a passo, pelo autor.

As páginas do Toolbook são construídas através de um editor gráfico que possui mecanismos especiais para desenho e montagem, através do posicionamento espacial dos diversos componentes de diversos tipos de mídia.

Um livro Toolbook pode ser navegado, simplesmente passando-se as páginas, ou por intermédio de elos de hipermídia que podem ser atribuídos aos componentes de uma página, ou a uma página inteira. As ações de um elo são definidas utilizando-se uma linguagem de programação script, a linguagem OpenScript. Essas ações podem ser atribuídas a eventos quaisquer, por exemplo, ao pressionar um botão, na exibição de um determinado componente, ou até no início da apresentação de um livro.

### **2.6.1 Interface com o Usuário**

A interface do sistema Toolbook assemelha-se à interface de um editor gráfico que permite o posicionamento de vários objetos de mídias diferentes na tela. As páginas de um livro Toolbook são constituídas pelas partes frontal e pelo background. Os objetos podem ser posicionados livremente dentro de qualquer parte de uma página. A Figura 2.10 mostra a interface do Toolbook e ilustra um exemplo de uma página de um livro.



Figura 2.9: A interface gráfica do sistema Toolbook

A razão de separar a parte frontal de uma página, é permitir que um mesmo background seja utilizado em várias páginas de um livro. Na interface, o autor dispõe de um conjunto de ferramentas que auxiliam na montagem do seu livro (documento hipermídia).

A interface do Toolbook possui dois modos de operação:

- modo autor;
- e modo leitor.

No modo autor, o usuário pode construir um novo livro, ou alterar um livro já construído. Já no modo leitor, o usuário se restringe às operações de leitura. Os modos de operação da interface do sistema Toolbook possuem funções análogas às funções dos autores e leitores de livros convencionais.

O Toolbook não possui um editor de sincronismo, não sendo possível definir os aspectos de sincronização temporal entre objetos no domínio do tempo.

## 2.7 Director

O *Director* é um sistema hipermídia criado pela MacroMind Corporation dos Estados Unidos. O sistema Director é baseado na metáfora de teatro. Seguindo essa metáfora, o autor de um documento possui estágios, como a construção de cenas, para montar a apresentação de um hiperdocumento [MacD89].

Para editar a sincronização temporal de um documento, o Director utiliza um editor de timeline, onde o usuário monta a seqüência de ações diretamente no tempo, selecionando as transições entre as cenas.

O sistema Director também possui uma linguagem *scriptware* chamada “Lingo” que permite definir relações de sincronização mais elaboradas.

O Director possui ainda editores gráficos sofisticados com vários recursos para a edição de das mídias texto e imagem gráfica, nesse último incluindo a construção de seqüências temporais de apresentação de imagens.

## 2.8 MAEstro

O sistema de autoria multimídia *MAEstro* foi inicialmente concebido por um grupo de pesquisas de informações acadêmicas da *Stanford University* sendo o primeiro protótipo implementado para a plataforma UNIX. Seu objetivo foi simplificar o processo de criação de documentos multimídia. Os principais fatores que influenciaram a arquitetura do MAEstro foram [DrGr91] [DrGr93]:

- a capacidade de incluir um fragmento de informação, codificado em exibidores de mídia quaisquer, em um documento multimídia. Na implementação existente, qualquer aplicação UNIX pode ser, potencialmente, um exibidor MAEstro;
- o foco ao processo de autoria de documentos baseado no paradigma de timeline;
- a capacidade de adicionar segmentos de mídia de maneira rápida e fácil.

O modelo de sincronização temporal do sistema MAEstro baseia-se no posicionamento dos segmentos de mídia diretamente em um eixo do tempo, sem possuir qualquer relacionamento com outros segmentos.

Um relógio é disparado no início da apresentação de um documento multimídia. Quando esse relógio alcança o momento de exibição de um novo segmento de mídia, é enviada uma mensagem para o exibidor apropriado executar sua apresentação. Após o início da exibição, o relógio interno é comparado com o relógio real do computador e, posteriormente, atualizado. Em alguns exibidores de mídia, esse processo pode causar um “salto” de alguns segundos no relógio do sistema. Se outro segmento de mídia foi programado para iniciar exatamente nesse tempo de “salto”, o sistema simplesmente deixa de exibi-lo.

Como os segmentos de mídia, em geral, estão posicionados em instantes diferentes no tempo, esse processo pode causar muitos “saltos” durante a apresentação de um documento multimídia. Em alguns casos, os autores podem rearrumar os segmentos de mídia no eixo do tempo para tentar evitar esse tipo de problema, porém isso pode não ser possível em todos os casos, além de que esse processo de rearrumação manual de todos os segmentos de mídia é um processo tedioso e não-confiável.

O sistema MAEstro utiliza um esquema de sincronização temporal onde apenas os pontos iniciais dos segmentos de mídia são sincronizados no tempo. Segundo esse esquema, o início da apresentação de todos os segmentos de mídia é previsível, porém não se pode prever o término de suas exibições, nem se pode afirmar como esses segmentos manterão a sincronização com os outros segmentos de mídia do documento. Isto é, apesar dos inícios das apresentações dos componentes de um documento serem necessariamente síncronos, não há garantia de que a apresentação do documento por completo será síncrona.

### **2.8.1 Interface com o Usuário**

A interface do sistema MAEstro para a construção de documentos multimídia é constituída pelo editor de timeline, ilustrado na Figura 2.10 [DrGr91] [DrGr93].

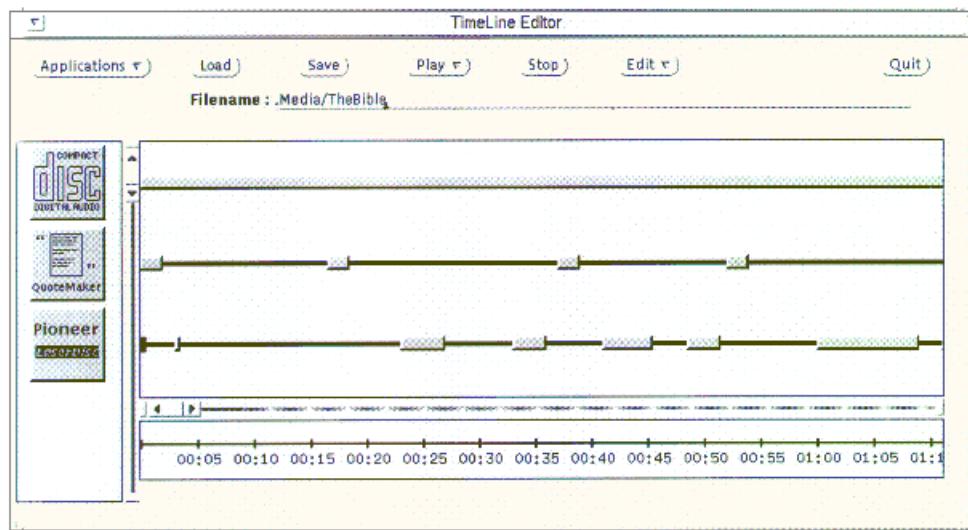


Figura 2.10: O Editor de Timeline do sistema MAEstro

A aplicação editor de timeline, representa um documento como um conjunto de trilhas (*tracks*) de tempo. Cada trilha corresponde a um determinado exibidor de mídia do sistema (ver Figura 2.10). O modelo da interface utilizada é simples e funciona bem para pequenas apresentações que não possuam interação do usuário.

O editor de timeline controla as ações sobre os segmentos de mídia, enviando mensagens para os exibidores das mídias nos momentos de exibição dos segmentos.

# Capítulo 3

## O Sistema HyperProp

### 3.1 Introdução

As aplicações multimídia/hipermídia foram construídas para atender requisitos específicos de uma determinada área de conhecimento. Em geral, essas aplicações são desprovidas de mecanismos que forneçam compatibilidade com outras aplicações, além de serem baseadas em plataformas de hardware e software específicas. Essas características tornam inviável o processo de reutilização dos objetos de dados que são manipulados por essas aplicações.

Um sistema hipermídia deve fornecer mecanismos que possibilitem a reutilização de objetos, dado que o armazenamento desses objetos, muitas vezes, representa um investimento significativo.

Diferente das características adotadas pelos diversos sistemas hipermídia, que na sua grande maioria visavam construir sistemas monolíticos e auto-suficientes, o sistema HyperProp, objeto desse capítulo, foi idealizado baseando-se no fato de que os objetos multimídia/hipermídia existentes devem ser reutilizados.

O HyperProp é constituído de um modelo conceitual de dados, o *Modelo de Contextos Aninhados* (MCA), e de uma arquitetura aberta cujo esquema de interfaces visa separar os componentes de dados dos componentes de apresentação, com o objetivo de tornar o HyperProp um sistema portável [Soar95]. Por se tratar de um sistema

aberto, uma aplicação poderá, dependendo de sua necessidade, utilizar o HyperProp através de qualquer nível de interface, como será visto em detalhes mais adiante.

Uma das principais características do sistema HyperProp é sua conformidade com a proposta de padrão MHEG, permitindo que seus objetos de dados possam ser intercambiados com outras aplicações que também sigam o padrão, além de permitir a reutilização direta de objetos MHEG existentes.

O sistema HyperProp possui ferramentas especiais para edição da estrutura e sincronizações temporal e espacial de documentos, que visam facilitar o processo de autoria, oferecendo um ambiente completo para manipulação de documentos. A utilização de editores gráficos para auxiliar no processo de autoria é crucial para qualquer sistema hipermídia. O EBS, foco principal dessa dissertação, é uma ferramenta do sistema HyperProp que possui a função de edição do sincronismo temporal e espacial em documentos multimídia/hipermídia.

A arquitetura do sistema HyperProp será apresentada na Seção 3.2, onde será descrita sua estrutura em camadas. Na Seção 3.3 é apresentado o Modelo de Contextos Aninhados, modelo conceitual de dados do HyperProp, onde suas principais entidades serão descritas baseando-se na hierarquia de classes do modelo. A Seção 3.4 fica reservada para explorar detalhes relacionados com os aspectos de apresentação de documentos no MCA, incluindo a definição das sincronizações temporal e espacial.

### 3.2 Arquitetura do HyperProp

Nesta seção é apresentada a arquitetura do sistema HyperProp [SoCC93], cuja estrutura possui três camadas, conforme pode ser visualizado na Figura 3.1.

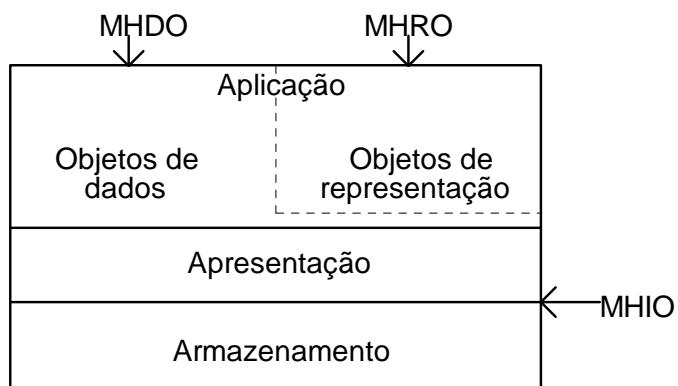


Figura 3.1: Estrutura em camadas do sistema HyperProp

A *camada de armazenamento* visa fornecer mecanismos para o armazenamento persistente de objetos multimídia/hipermídia. Ela é responsável, entre outras coisas, pela organização dos dados segundo a hierarquia de classes do modelo conceitual utilizado, o Modelo de Contextos Aninhados.

A interface entre a camada de armazenamento e a camada de apresentação é chamada de MHIO (*multimedia hypermedia interchangeable objects interface*). A interface MHIO é a chave para o fornecimento de compatibilidade entre aplicações e plataformas de exibição. São dois os pontos com os quais as duas camadas devem concordar:

- a representação dos objetos multimídia/hipermídia a serem intercambiados deve estar em conformidade com a proposta de padrão ISO MHEG;
- a equivalência entre as mensagens, requisições e confirmações utilizadas nessas camadas.

A *camada de aplicação* incorpora dois conceitos para os objetos do sistema HyperProp, os *objetos de dados* e os *objetos de representação*. Conceitos similares podem ser encontrados em [PuGu90]. Um *objeto de dados* é criado ou como um objeto totalmente novo, ou como uma cópia local de um objeto de armazenamento, acrescida de novos atributos (não-persistentes) que são dependentes da aplicação. Ele contém métodos para manipular os novos atributos, assim como métodos para manipular a informação originalmente pertencente ao objeto de armazenamento, com exceção do atributo conteúdo que não é interpretável nesse nível de abstração. A classe de *objeto de representação* é criada a partir da classe de objeto de dados, adicionando-se novos métodos para exibir e editar o atributo conteúdo no formato mais apropriado para um uso particular da informação. O EBS atuará, como será discutido mais adiante neste capítulo, no nível dos objetos de representação. Nesta camada, as aplicações podem fazer uso das interfaces MHRO (*Multimedia*

*Hypermedia Representation Object*) ou MHDO (*Multimedia Hypermedia Data Object*), além da já mencionada MHIO.

A *camada de apresentação* não implementa nenhum método associado aos objetos de dados. Sua função é converter o formato dos objetos de dados, padrão definido pela interface MHIO, para o formato de dados dos objetos usados por uma aplicação e plataforma particular, e vice versa.

A implementação do HyperProp segue uma arquitetura baseada no modelo cliente/servidor, conforme ilustrado na Figura 3.2, onde o servidor é distribuído. Nas estações clientes o nível de aplicação e de apresentação funcionam exatamente como mencionado anteriormente, com uma pequena diferença, o nível de apresentação deverá também ser capaz de desmontar e remontar os objetos multimídia especificados pela interface MHIO.

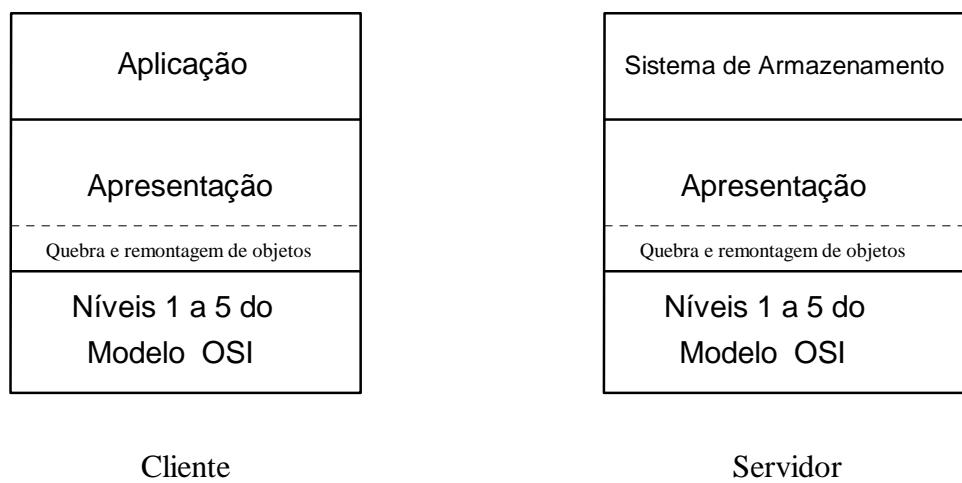


Figura 3.2: Arquitetura cliente/servidor do HyperProp

### 3.3 O Modelo de Contextos Aninhados

A definição de documentos hipermídia no *Modelo de Contextos Aninhados* (MCA), modelo conceitual do sistema HyperProp, se baseia nos conceitos usuais de *nós* e *elos*. *Nós* são fragmentos de informação. O modelo distingue duas classes básicas de nós, chamados de *nós terminais* e *nós de composição*, sendo os nós de composição o ponto central do modelo. Um *nó de composição* é um tipo de *nó* cujo conteúdo são *nós* e *elos*, incluindo *nós de composição*, de maneira recursiva. No MCA, os *elos* são

definidos dentro dos *nós de composição* e permitem a definição de relacionamentos entre *nós* componentes, incluindo as relações de sincronismo. A Figura 3.3 mostra a hierarquia de classes do MCA.

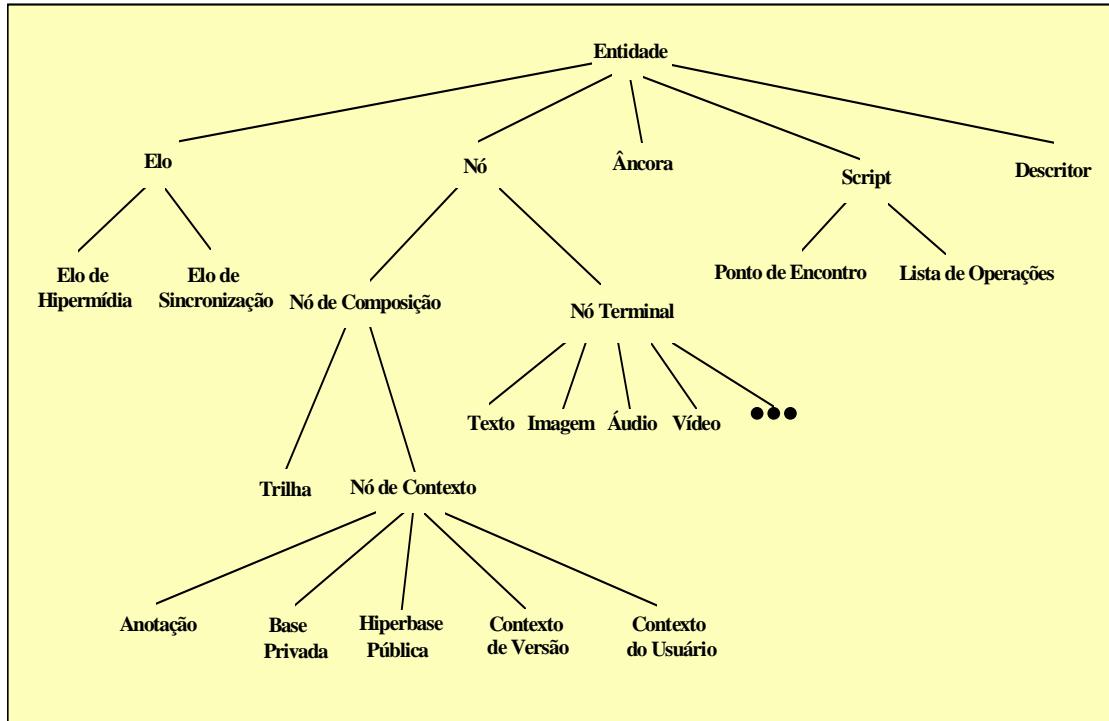


Figura 3.3: A Hierarquia de Classes do MCA

Uma *entidade* é uma classe que possui como atributos, um identificador único e uma *lista de controle de acesso*. Para cada entidade, a lista de controle de acesso identifica um usuário, ou um grupo, e seus direitos de acesso sobre cada atributo da entidade.

Um *nó* possui como atributos adicionais um *conteúdo* e um *conjunto de âncoras*. Uma *âncora* é definida por uma região específica de um nó. O conjunto de âncoras de um nó funciona como uma máscara que é interpretada como uma interface externa do nó, ou seja, o conjunto de âncoras definem os pontos onde o nó pode ser acessado externamente.

A classe de *nós terminais* pode ser especializada em outras classes (nó de texto, nó de imagem gráfica, nó de áudio, nó de vídeo, etc.), conforme requisitado pelas aplicações.

Intuitivamente, um *nó terminal* contém dados cuja estrutura interna é codificada de acordo com a aplicação e é independente do modelo.

Os *nós de contexto* são especializações dos nós de composição e contém um conjunto de nós, terminais ou de contexto, recursivamente. Nós de contexto são usados para estruturar documentos, para definir visões diferentes do mesmo documento, nos mecanismos de controle de versão e sincronização, e na definição de bases de dados públicas e privadas.

O MCA permite que um mesmo nó esteja contido em diferentes nós de composição, e que o aninhamento desses nós seja feito em qualquer nível de profundidade. A *perspectiva* de um nó define qual o caminho de aninhamento que um nó está contido. Uma perspectiva para um nó  $N$  é uma seqüência  $P=(N_m, \dots, N_1)$ ,  $m \geq 1$ , tal que  $N = N_1$ ,  $N_{i+1}$  é o nó de composição imediatamente superior na hierarquia que contenha  $N$ , para  $i \in [1, m]$ , e  $N_m$  é o nó de composição mais externo em nível de aninhamento que corresponde à base privada do usuário. Um nó pode possuir diferentes perspectivas, isto é, ele pode estar contido em vários nós de composição ao mesmo tempo. O exemplo ilustrado pela Figura 3.4 mostra o aninhamento estrutural dos nós terminais A, B, C, D e E, e dos nós de contexto C1, C2, e C3 de acordo com o Modelo de Contextos Aninhados, onde o nó D encontra-se contido nos nós de contexto C1 e C3, e portanto, com perspectivas diferentes.

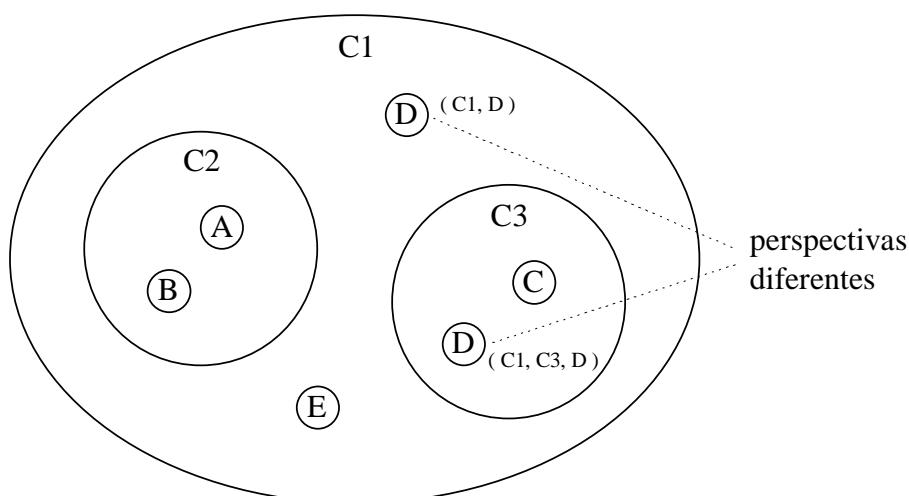


Figura 3.4: Nô D com perspectivas diferentes

Os *elos* são caracterizados como uma relação (m:n) entre dois conjuntos de *eventos* (origem e destino) associados a um ou mais *nós*. O *elo* possui um atributo especial, chamado *ponto de encontro*. Nesse atributo são especificadas as ações, a serem executadas sobre as extremidades de destino do elo, e a relação entre os eventos origem que funciona como a *condição de disparo* das ações do elo. Dependendo da natureza dos eventos origem do elo, ele pode ser classificado como *elo de hipermídia* ou como *elo de sincronização*. Eventos, ações e elos são conceitos importantes para a definição das relações de sincronização temporal, e serão abordados em detalhes na Seção 3.4.

Um objeto *descritor* pode ser associado a várias entidades. Ele contém informação que determina como a entidade deve ser apresentada. As informações de alteração de comportamento são definidas dentro dos descritores. Elas definem, por exemplo, como será realizada a sincronização espacial de um determinado nó. Os objetos descritores serão analisados detalhadamente na Seção 3.4.3.

A *hiperbase pública*, subclasse especial do nó de contexto, corresponde a um repositório de informações públicas e estáveis, enquanto que a *base privada* corresponde a uma sessão de trabalho do usuário e contém entidades que estão em uso durante a sessão.

Uma discussão detalhada sobre o modelo de contextos aninhados pode ser encontrada em [Soar95, SoSC95, Bati94, SoCR94 e SoCa93], onde também são apresentadas as noções de estruturas virtuais, controle de versão e de trabalho cooperativo, onde as classes *anotação* e *contexto de versões*, são introduzidas.

### 3.4 O Subsistema de Apresentação do HyperProp

O modelo conceitual estrutural básico do MCA trata um hiperdocumento como uma estrutura de dados passiva. Um sistema hipermídia deve, no entanto, fornecer ferramentas para que o usuário possa ter acesso à estrutura da rede de interconexões, exibi-la e definir a sincronização temporal e espacial entre as entidades. No HyperProp essa funcionalidade é exercida pelos componentes do subsistema de apresentação que serão analisados em maiores detalhes nas subseções 3.4.1 a 3.4.4 [SoSC95] [SoSo96].

O EBS, ferramenta do sistema HyperProp, utiliza diretamente os componentes do subsistema de apresentação do MCA para construir as visões temporal e espacial de um documento multimídia/hipermídia. A Subseção 3.4.5 mostra como operações típicas de sincronização temporal são modeladas utilizando-se os componentes do subsistema de apresentação.

Uma outra ferramenta do sistema HyperProp, o Browser de Base Privada, é apresentado na Subseção 3.4.6. Essa ferramenta fornece a visão estrutural dos documentos contidos numa seção do usuário. Sua utilização, juntamente com o EBS, fornecerá uma ferramenta completa para a visualização e manipulação de um documento sob as perspectivas estrutural, temporal e espacial. O EBS e o browser de base privada são métodos da entidade *base privada* do Modelo de Contextos Aninhados.

### 3.4.1 Eventos e Ações

O subsistema de apresentação do HyperProp permite que sejam associados estados às regiões dos nós. Por exemplo, um *estado* indica se uma região de um nó está sendo selecionada (estado de seleção) ou sendo exibida (estado de apresentação).

Um evento é um acontecimento atômico caracterizado por mudanças nos estados das regiões dos nós. Existem três tipos de eventos:

- *start-presentation*
- *end-presentation*
- *selection*

O evento de início de apresentação de uma região (*start-presentation*) ocorre quando uma região inicia a sua exibição para o usuário. O evento de fim de apresentação de uma região (*end-presentation*) ocorre quando uma região termina de ser exibida para o usuário. O evento de seleção de uma região (*selection*) ocorre quando uma região é selecionada pelo usuário.

Os eventos de um nó ocorrem seguindo uma seqüência temporal especificada ou pelo próprio tempo de exibição do conteúdo do nó, ou explicitamente pelo usuário. A região especificada por um evento pode ser definida por uma âncora, no caso da definição de elos. A relação entre os eventos e os elos é mostrada na próxima subseção.

No caso de mídias contínuas (áudio e vídeo), o tempo de exibição de um nó pode ser igual ao tempo de exibição do próprio conteúdo desse nó, enquanto que o tempo de exibição de um nó de uma mídia estática (texto e imagem gráfica) pode ser definido pelo usuário durante o processo de especificação da duração temporal do nó. A Figura 3.5 mostra um nó A no domínio do tempo e as relações entre suas regiões e eventos.

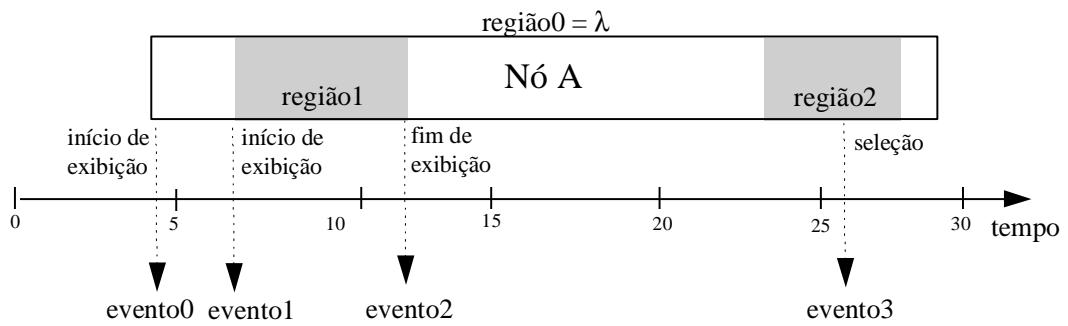


Figura 3.5: Relação entre as regiões e eventos do nó A

No exemplo da Figura 3.5, o nó A possui as regiões *região1* e *região2*, que foram previamente definidas pelo usuário, e a região *região0*, representada pelo símbolo  $\lambda$  que corresponde ao nó A inteiro. É importante ressaltar que todo nó possui uma região que corresponde ao nó inteiro, representada pelo símbolo  $\lambda$ . A seleção da região *região2* gera o evento *evento3* (*selection*), enquanto que a exibição da região *região1* gera dois eventos, o evento *evento1* que ocorre no início da exibição (*start-presentation*), e o evento *evento2* que ocorre no fim da exibição da região (*end-presentation*). O evento *evento0* ocorre no início da exibição da região *região0*.

Os eventos são importantes para a definição de relacionamentos entre os nós de um documento hipermídia, pois são os pontos de conexão dos elos, conforme será visto na subseção seguinte.

As ações correspondem a métodos que podem ser atribuídos a um nó inteiro, ou a uma região de um nó. A relação entre uma ação e uma região de um nó pode ser 1:1, ou seja, quando existe uma ação para cada região de um nó, ou 1:n, quando a mesma ação pode ser atribuída a várias regiões de nós diferentes. A tabela 3.1 mostra as ações, seus significados e seus tipos de relações.

<i>Ações</i>	<i>Significado</i>
<b>play (ou run)</b> <b>(relação 1:1)</b>	iniciar a exibição de uma região de um nó
<b>play sync</b> <b>(relação 1:n)</b>	iniciar a exibição síncrona de duas ou mais regiões de nós ( ver detalhes na Subseção 3.4.5 )
<b>stop</b> <b>(relação 1:1)</b>	encerrar a exibição de uma região de um nó
<b>pause</b> <b>(relação 1:1)</b>	dar uma pausa na exibição de uma região de um nó
<b>resume</b> <b>(relação 1:1)</b>	reiniciar a exibição de uma região de um nó

Tabela 3.1: Ações que podem ser atribuídas a uma região de um nó

### 3.4.2 Elos

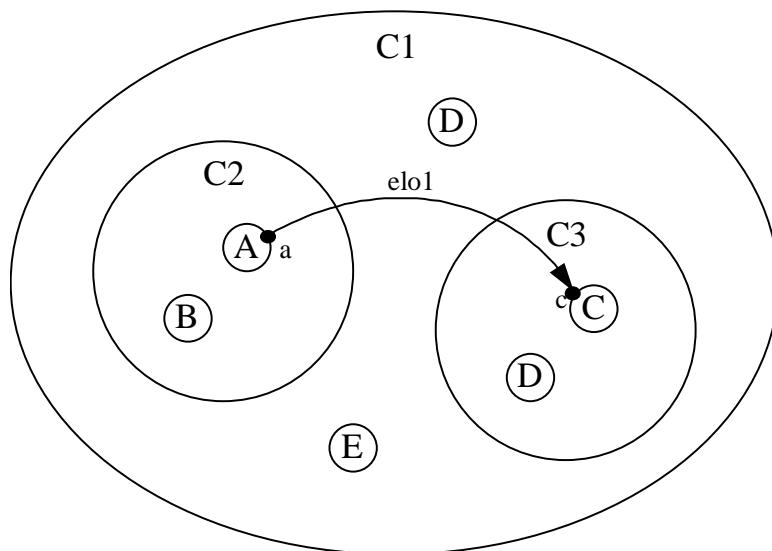
No subsistema de apresentação do MCA, os elos são interpretados como conexões lógicas (relação n:m) entre conjuntos de eventos.

Um elo é uma entidade que possui como atributos adicionais um conjunto de *pontos terminais origem*, um conjunto de *pontos terminais destino*, um objeto ponto de encontro e descritores.

Um *ponto terminal* de um elo é definido como um par na forma  $\langle (N_k, \dots, N_2, N_1), A \rangle$  onde  $N_{i+1}$  é um nó de composição, e  $N_i$  está contido em  $N_{i+1}$ , para todo  $i \in [i, k)$ , com  $k$

$> 0$ , e  $A$  é uma âncora de  $N_I$ . Cada ponto terminal origem de um elo especifica um evento na âncora desse ponto terminal. Para isso, é adicionado ao ponto terminal origem, uma *condição de ocorrência* da âncora. As condições de ocorrência podem ser: *exibição* ou *seleção*. Quando um evento especifica o início da apresentação de uma âncora, utiliza-se a condição de ocorrência *início de exibição*, quando um evento especifica o fim da apresentação de uma âncora, utiliza-se a condição de ocorrência *fim de exibição*, quando um evento especifica a seleção do usuário em uma âncora, usa-se a condição de ocorrência *seleção*. As ações são adicionadas aos pontos terminais destino do elo para especificar as ações que devem ser executadas pelo elo.

A Figura 3.6 mostra a visão estrutural e a especificação de um elo *elo1* que interconecta o evento gerado pela seleção da âncora  $a$  do nó  $A$  com a ação de início da exibição da âncora  $c$  do nó  $C$ .



ponto terminal origem: { PT1 =  $\langle (C1, C2, A), a \rangle$  }

evento: { PT1, seleção }

ponto terminal destino: { PT2 =  $\langle (C1, C3, C), c \rangle$  }

ação: { PT2, play }

Figura 3.6: Exemplo de um elo com relação (1:1)

A relação associada a um elo é expressa através do seu atributo ponto de encontro. O *ponto de encontro* consiste de uma expressão lógica, composta por um conjunto de condições e um conjunto de ações. As condições são baseadas nos pontos terminais origem do elo, enquanto que as ações baseiam-se nos pontos terminais destino do elo.

Quando a expressão lógica definida pelo ponto de encontro é satisfeita, ela dispara as ações associadas aos pontos terminais destino do elo. A Figura 3.7 mostra a relação de um elo (n:m) genérico composto por  $n$  eventos origem,  $m$  ações destino, e um ponto de encontro que define a condição de disparo do elo.

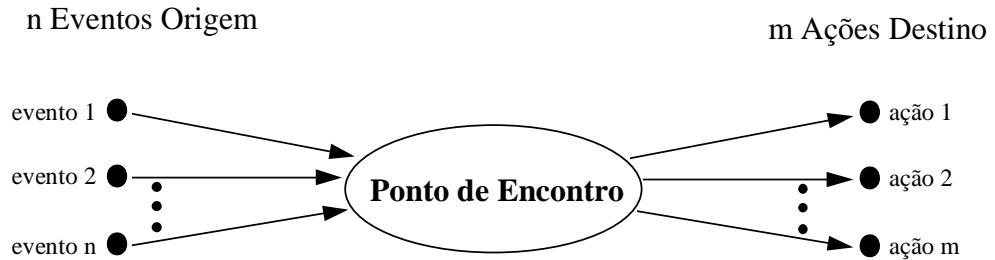


Figura 3.7: Relação entre os eventos e as ações de um elo (n:m)

Os operadores que podem ser utilizados para a construção de uma expressão lógica de um ponto de encontro são mostrados na Tabela 3.2.

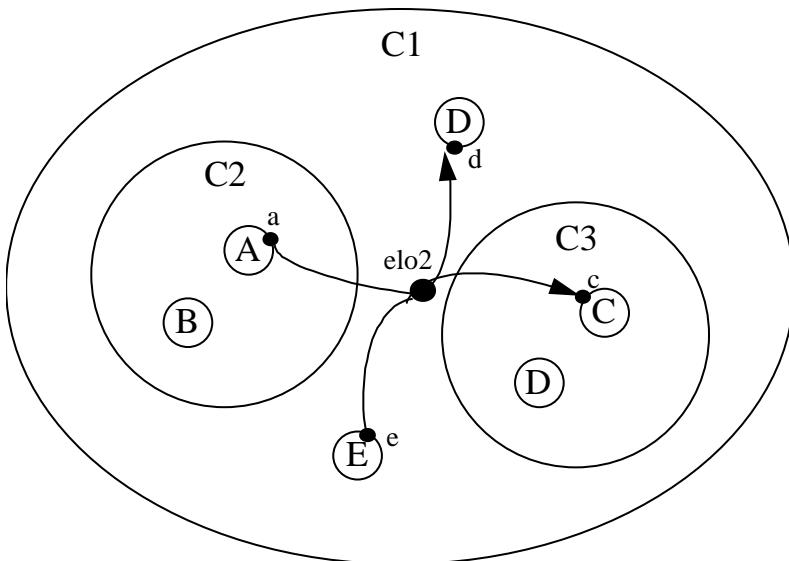
<i>Operador</i>	<i>Significado</i>
$\wedge$	o mesmo que o operador and convencional
$\neg$	o mesmo que o operador not convencional
$ $	o mesmo que o operador or convencional
$( )$	serve para agrupar sentenças
$\oplus n \text{ seg}$	define um valor de retardo (delay)

Tabela 3.2: Operadores de um ponto de encontro

Na Subseção 3.4.5 será mostrado que, com os elos do MCA e com os operadores do ponto de encontro mostrados na Tabela 3.2, é possível construir todas as relações temporais definidas por Allen [Alle83] [Alle84] [AlFe94].

A Figura 3.8 mostra o exemplo de um elo *elo2* que é uma relação (2:2). Nessa figura é mostrado o ponto de encontro e suas condições e ações definidas pela expressão lógica.

O atributo ponto de encontro do elo *elo2* da Figura 3.8 pode ser descrito da seguinte forma — ao acontecer o evento E1 e decorrerem 5.4 segundos, e ao ocorrer o evento E2 e decorrerem 4.6 segundos, exiba o ponto terminal PT3 e pare a exibição do nó especificado pelo ponto terminal PT4.



pontos terminais origem: { PT1 = <(C1, C2, A), a> , PT2 = <(C1, E), e> }  
 eventos: { E1 = (PT1, seleção) , E2 = (PT2, exibição) }  
 pontos terminais destino: { PT3 = <(C1, C3, C), c> , PT4 = <(C1, D), d> }  
 ponto de encontro: ( E1 + 5.4 seg) ^ (E2 + 4.6 seg) => play( PT3 ), stop( PT4 )

Figura 3.8: Exemplo da estrutura e definição de um elo (2:2)

Dependendo das condições de ocorrência dos eventos que constituem a condição de disparo de um elo, ele pode ser classificado em dois tipos:

- *elo de hipermídia*;
- ou *elo de sincronização*.

Se existir pelo menos uma condição de ocorrência do tipo *seleção* em pelo menos um dos eventos da expressão lógica do elo, ele é considerado um *elo de hipermídia*. Caso contrário, ou seja, todas as condições de ocorrência de todos os eventos que constituem a condição de disparo do elo forem do tipo *exibição*, o elo é considerado um *elo de sincronização*.

### 3.4.3 Descritores

Um *descritor* especifica a maneira como um objeto de dados será exibido para o usuário, detalhando como ele será iniciado, qual dispositivo de E/S será utilizado e quais mudanças ocorrerão no seu comportamento durante a exibição, caso elas existam. As mudanças de comportamento são programadas como listas de operações, como em [BuZe92]. Um descritor consiste de uma especificação de inicialização e uma coleção de descrições de eventos.

A especificação de inicialização de um descritor contém a informação necessária para iniciar a apresentação de uma entidade, incluindo a definição dos métodos (exibidores) que serão utilizados para apresentá-la em um canal virtual específico. Um método pode ser qualquer programa exibidor de um tipo de mídia. Os canais virtuais identificam os dispositivos de exibição (display ou áudio) que serão utilizados para apresentar um nó, e dependendo do tipo de mídia a ser exibido, os canais definem a posição na tela e a dimensão da janela de exibição (canal de display), ou o volume de exibição (canal de áudio). A especificação de inicialização também possui uma lista de operações que são executadas antes do nó ser apresentado.

Cada descrição de evento de um descritor pode definir um ponto de sincronização espacial, onde o comportamento da exibição de um nó terminal ou de composição pode ser alterado. Um evento de mudança de comportamento contém a descrição de uma região, uma especificação da condição de ocorrência do evento e uma lista de operações a serem executadas. A lista de operações consiste de um conjunto de condições e ações, onde as condições devem ser satisfeitas para disparar as ações. A Figura 3.9 mostra a especificação de dois descritores *desc1* e *desc2*, que podem ser associados a dois nós quaisquer.

<p>DESCRIPTOR <i>desc1</i>: <b>start-up</b>- <i>action</i> play  <i>exibitor</i> windows word  <i>channel</i> display  <i>position</i> &lt;50, 100&gt;  <i>size</i>: &lt;60, 150&gt;  <i>duration</i> 15 seconds</p> <p><b>events</b>- ( anchor1, exhibition, delay = 0 ) <math>\Rightarrow</math> ( <i>size</i> &lt;60, 200&gt; )  ( <math>\lambda</math>, exhibition, delay = 10 ) <math>\Rightarrow</math> ( <i>position</i> &lt;60, 100&gt; )</p>
<p>DESCRIPTOR <i>desc2</i>: <b>start-up</b>- <i>action</i> play  <i>exibitor</i> audio tool  <i>channel</i> audio  <i>volume</i> 100%  <i>duration</i> 34.7 seconds</p> <p><b>events</b>- ( anchor4, exhibition, delay = 0 ) <math>\Rightarrow</math> ( <i>volume</i> 70% )  ( <math>\lambda</math>, exhibition, delay = 30 ) <math>\Rightarrow</math> ( <i>volume</i> 50% )</p>

Figura 3.9: Exemplos da especificação de descritores

A especificação de inicialização (*start-up*) do descritor *desc1* mostrado na Figura 3.9, define que o nó a ele associado deve ser apresentado pelo exibidor *windows word*, utilizando o canal *display*, na posição de tela <50, 100> (que especifica o canto superior esquerdo da janela de exibição em pixels), cuja dimensão da janela é de <60, 150> pixels, e com duração de 15 segundos. A lista de descrições de eventos do descritor *desc1* define que o evento ocasionado pela exibição da âncora *anchor1* acarretará a mudança da dimensão da janela de exibição para <60,200>, e o evento ocasionado pela exibição da âncora  $\lambda$ , adicionado de um retardo de 10 segundos, irá alterar a posição da janela de exibição para <60,100> (canto superior esquerdo da janela).

### 3.4.4 Objetos de Representação

A informação necessária para apresentar os componentes de um documento multimídia é fornecida pelo objeto de representação. Os *objetos de representação* são criados em tempo de execução e são responsáveis pela detecção e sinalização da ocorrência de eventos. Um *objeto de representação* é formado pela associação de um descritor e um objeto de dados [SoSC95] [SoSo96]. Um descritor pode ser associado a um nó de três maneiras diferentes:

- um nó pode conter um objeto descritor como atributo, definido explicitamente pelo usuário;
- um descritor pode ser associado a um nó através de um elo;
- um nó de contexto que contém um determinado nó pode associar um descritor a esse nó;
- ou então o descritor *default* da classe é associado ao nó.

A associação entre objetos de dados e descritores pode ser visualizada na Figura 3.10 por linhas conectando os objetos de dados no plano intermediário com os objetos de representação, desenhados no plano superior. Na figura, nós são representados por círculos, elos por arcos e composições pela inclusão de círculos e arcos em círculos maiores.

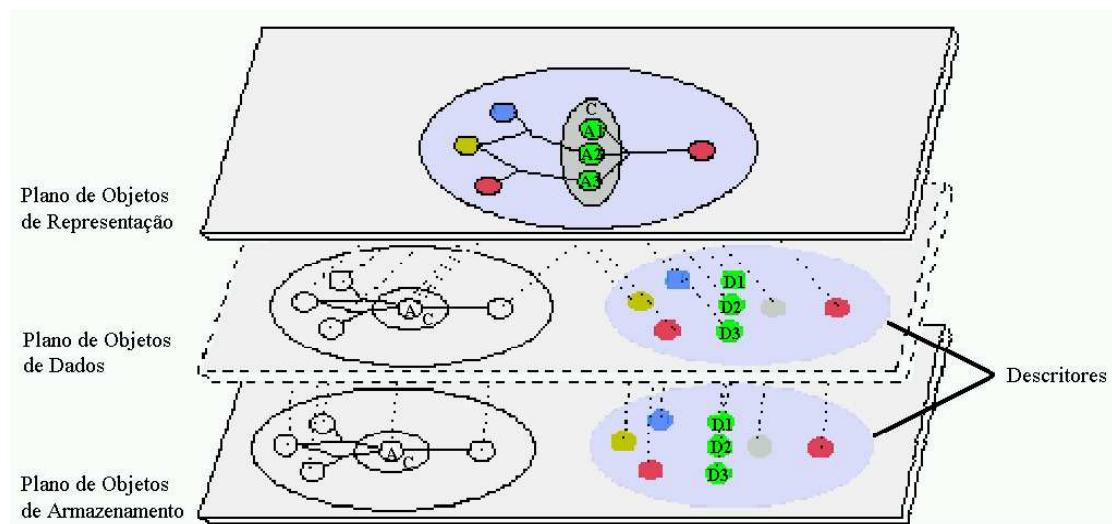


Figura 3.10: Planos de armazenamento, de dados e de representação

Note que o modelo de apresentação permite a combinação de uma entidade objeto de dados com diferentes descritores, originando diferentes representações (objetos de representação) da mesma entidade. A Figura 3.10 mostra essa característica com a associação dos descritores  $D_1$ ,  $D_2$  e  $D_3$  ao objeto de dados  $A$ , criando os objetos de representação  $A_1$ ,  $A_2$  e  $A_3$ . O nó  $A$  possui três diferentes representações porque existem três diferentes formas de navegação até ele, através dos dois elos ou através da hierarquia de composições.

Como existem diferentes representações para o mesmo nó, não é razoável manusear a estrutura de um documento no plano de representação. Isto certamente confundiria o usuário. Assim, o browser de base privada, a ser apresentado na Seção 3.4.6, atua no plano de objetos de dados, uma vez que toda a informação de estrutura dos documentos encontra-se nesse plano. Por outro lado, os relacionamentos de sincronismo possuem apenas significado no plano de representação. Assim, é nesse plano que o EBS atuará, como será descrito no Capítulo 4.

### 3.4.5 Operações de Sincronismo

O EBS oferece operações que permitem definir o sincronismo temporal entre dois ou mais objetos de representação através de relacionamentos de alto nível. Esses relacionamentos implicam na criação automática de elos de sincronização entre os nós envolvidos.

As tabelas que são apresentadas a seguir mostram as treze relações básicas da álgebra de intervalos definida por Allen [Alle83] [Alle84] [AlFe94]. Segundo Allen, qualquer relação temporal entre dois objetos baseia-se nessas treze relações básicas.

Nas figuras que são mostradas a seguir, essas relações são modeladas no domínio do tempo usando elos de sincronismo do MCA, que é a notação utilizada pelo EBS.

#### Relações 1 e 2:

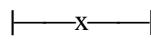
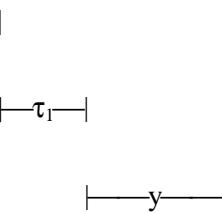
Relações	Significado	Elo MCA
x before y		
y after x		End-Presentation(x,λ) ⊕ τ <sub>1</sub> → Run(y,λ)

Tabela 3.3: Relações 1 e 2 da álgebra de intervalos de Allen

Essas relações são modeladas no MCA utilizando-se apenas um elo de sincronização com um retardo especificado no ponto de encontro de  $\tau_1$  segundos. A visualização do elo no domínio do tempo é mostrada na Figura 3.11.

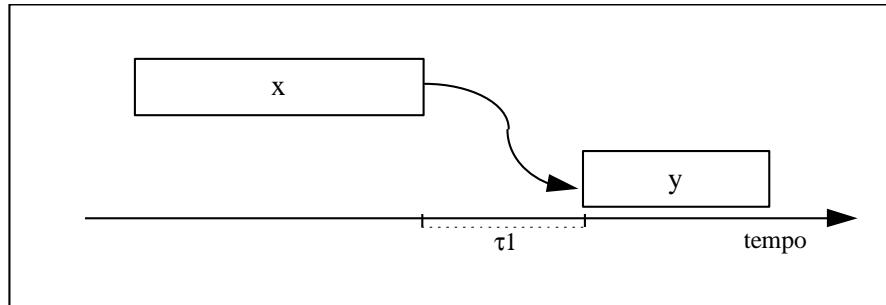


Figura 3.11: Relações 1 e 2 de Allen modeladas com um elo de sincronização

### Relações 3 e 4:

Relações	Significado	Elo MCA
x meets y	—x—	End-Presentation(x,λ) → Run(y,λ)
y met-by x	—y—	

Tabela 3.4: Relações 3 e 4 da álgebra de intervalos de Allen

Essas relações são modeladas no MCA utilizando-se apenas um elo de sincronização com retardo de zero segundos. A visualização da relação no domínio do tempo é mostrada na Figura 3.12.

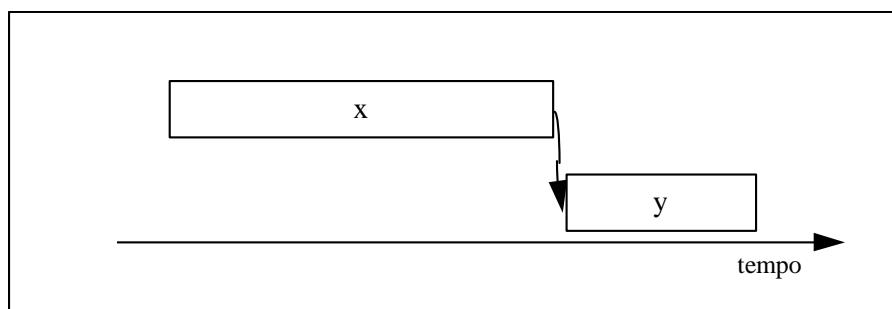


Figura 3.12: Relações 3 e 4 de Allen modeladas com um elo de sincronização

Relações 5 e 6:

Relações	Significado	Elo MCA
x overlaps y	—x—	
y overlapped-by x	—τ₂—   —y—	Start-Presentation(x,λ) $\oplus$ τ₂ $\rightarrow$ Run(y,λ)

Tabela 3.5: Relações 5 e 6 da álgebra de intervalos de Allen

Essas relações são modeladas no MCA utilizando-se apenas um elo de sincronização com um retardo de  $\tau_1$  segundos. A visualização da relação no domínio do tempo é mostrada na Figura 3.13.

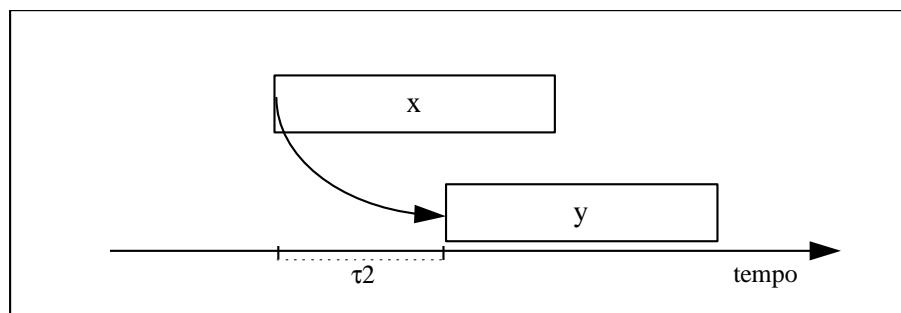


Figura 3.13: Relações 5 e 6 de Allen modeladas com um elo de sincronização

Relações 7 e 8:

Relações	Significado	Elos MCA
x during y	—x—	Start-Presentation(y,λ) $\oplus$ τ₃ $\rightarrow$ Run(x,λ)
y contains x	—τ₃—  $\quad$  —τ₄—   —y—	End-Presentation(x,λ) $\oplus$ τ₄ $\rightarrow$ Stop(y,λ)

Tabela 3.6: Relações 7 e 8 da álgebra de intervalos de Allen

Essas relações são modeladas no MCA utilizando-se dois elos de sincronização com retardos de  $\tau_3$  segundos e  $\tau_4$  segundos. A visualização da relação no domínio do tempo é mostrada na Figura 3.14.

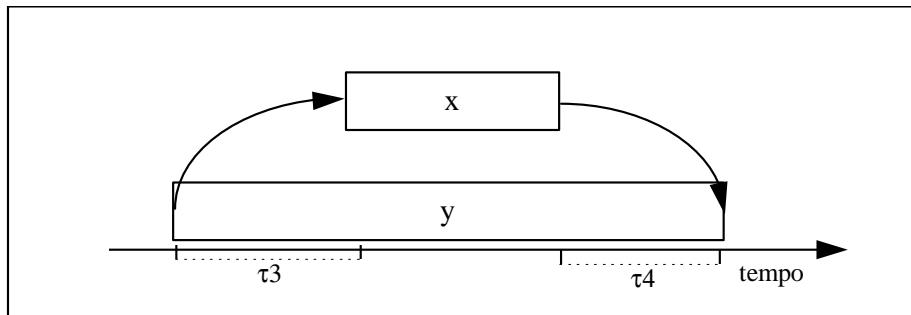


Figura 3.14: Relações 7 e 8 de Allen modeladas com elos de sincronização

Relações 9 e 10:

Relação	Significado	Elo MCA
x starts y	—x—	Start-Presentation(x,λ) → Run(y,λ)
y started-by x	—y—	

Tabela 3.7: Relações 9 e 10 da álgebra de intervalos de Allen

Essas relações são modeladas no MCA utilizando-se apenas um elo de sincronização com retardo de zero segundos. A visualização no domínio do tempo é mostrada na Figura 3.15.

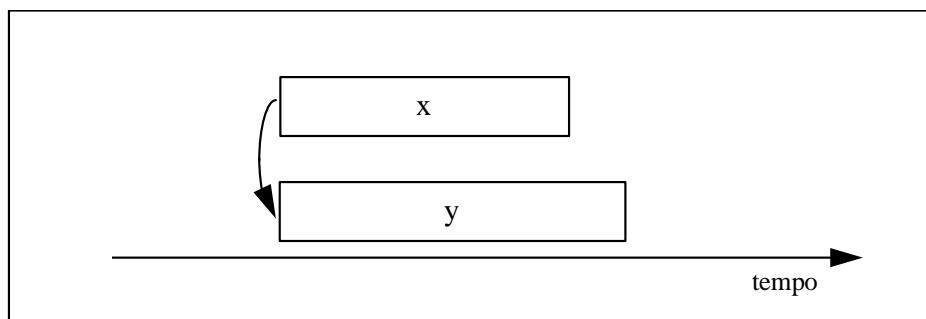


Figura 3.15: Relações 9 e 10 de Allen modeladas com um elo de sincronização

Relações 11 e 12:

Relação	Significado	Elo MCA
x finishes y	—x—	End-Presentation(x, $\lambda$ ) $\rightarrow$ Stop(y, $\lambda$ )
y finished-by x	—y—	

Tabela 3.8: Relações 11 e 12 da álgebra de intervalos de Allen

Essas relações são modeladas no MCA utilizando-se apenas um elo de sincronização com retardo de zero segundos. A visualização da relação no domínio do tempo é mostrada na Figura 3.16.

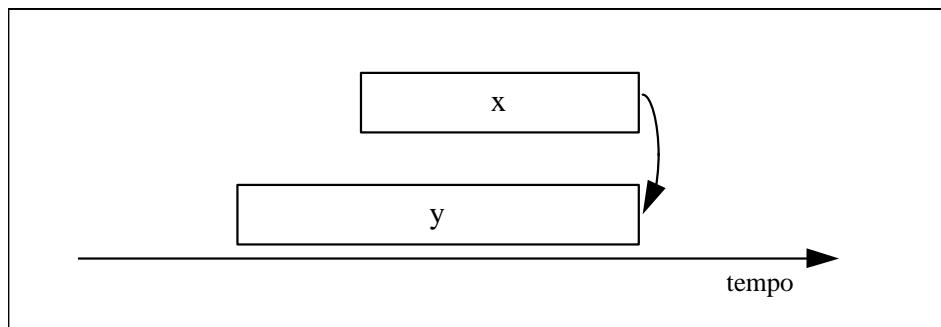


Figura 3.16: Relações 11 e 12 de Allen modelada com um elo de sincronização

Relação 13:

Relação	Significado	Elos MCA
x equal y	—x—	Start-Presentation(x, $\lambda$ ) $\rightarrow$ Run(y, $\lambda$ )
	—y—	End-Presentation(x, $\lambda$ ) $\rightarrow$ Stop(y, $\lambda$ )

Tabela 3.9: Relação 13 da álgebra de intervalos de Allen

Essa relação é modelada no MCA utilizando-se dois elos de sincronização ambos com retardos de zero segundos. A visualização da relação no domínio do tempo é mostrada na Figura 3.17.

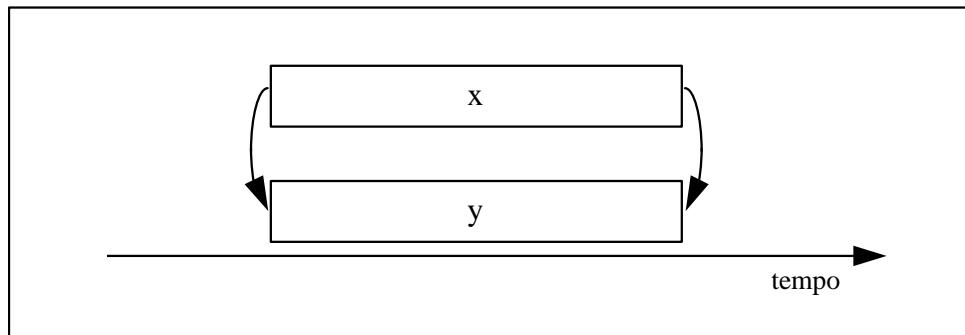


Figura 3.17: Relação 13 de Allen modelada com elos de sincronização

As relações (3 e 4), (9 e 10) e 13 da álgebra de intervalos de Allen são bastante utilizadas na definição de relações temporais e, por essa razão são chamadas de *relações de alto nível* pelo EBS . A Figura 3.18 mostra os elos criados pelo EBS para definir o sincronismo temporal dos objetos O1 e O2 segundo as três relações de alto nível descritas anteriormente.

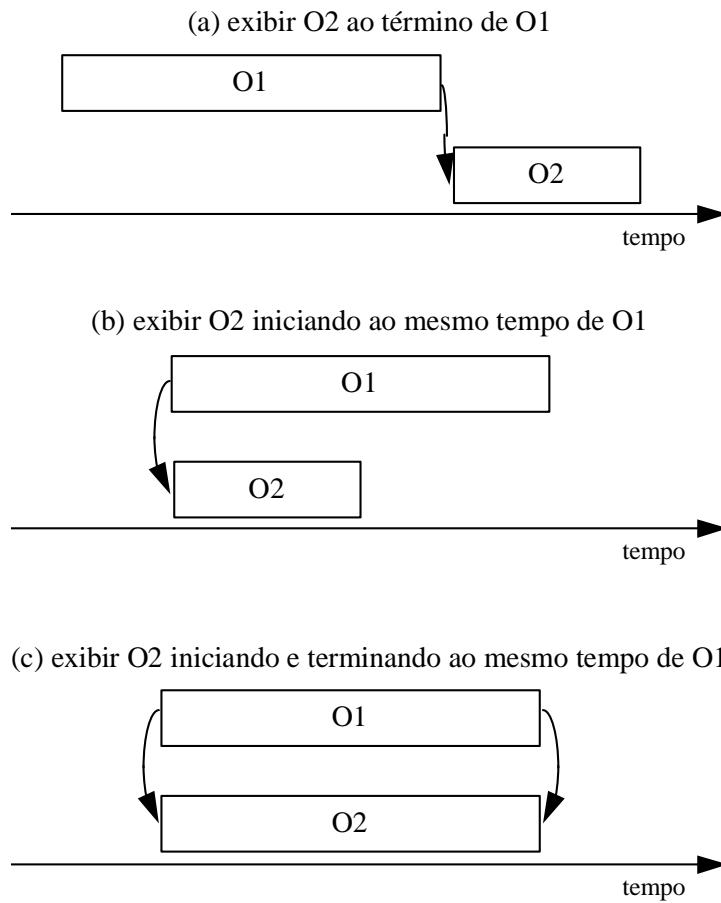


Figura 3.18: Relações de alto nível

Os objetos envolvidos nas operações de alto nível são classificados em duas categorias:

- *objeto base* da operação;
- e *objetos dependentes*.

Uma operação de alto nível sempre possui um objeto que é chamado de *objeto base* da operação, onde a relação com demais objetos da operação é definida baseando-se nele. No caso dos exemplos das relações mostradas pela Figura 3.18, o objeto base é o nó O1. Note que todos os elos envolvidos nas operações de alto nível da Figura 3.18 são relações (1:1), e os pontos terminais origem dos elos são eventos que ocorrem no objeto base da operação.

Os demais objetos envolvidos nas operações de alto nível são chamados de *objetos dependentes*, pois dependem do objeto base para que sejam apresentados. Os pontos terminais destino dos elos utilizados nas relações de alto nível são definidos nos

objetos dependentes. Nos exemplos da Figura 3.18, como as operações envolvem apenas dois nós, só existe um objeto dependente em cada operação, o nó O2.

A operação *exibir ao término de* é modelada utilizando-se um elo que conecta o evento de fim de exibição de um objeto com o evento de início da exibição do outro objeto, onde o valor do delay do atributo ponto de encontro desse elo é igual a zero. No exemplo da Figura 3.18.a, o nó O2 foi definido para ser exibido imediatamente após o término da exibição do nó O1.

A operação *exibir iniciando ao mesmo tempo* é definida por um elo que conecta os inícios das exibições de dois ou mais objetos, com o valor do delay do atributo ponto de encontro do elo sendo igual a zero. No exemplo da Figura 3.18.b, o nó O2 foi definido para ser exibido iniciando ao mesmo tempo do nó O1.

No caso da operação *exibir iniciando e terminando ao mesmo tempo*, um elo conecta os eventos que correspondem ao início da exibição dos nós, e outro elo conecta os eventos de fim de exibição dos nós. Ambos os elos devem possuir o valor do delay do atributo ponto de encontro igual a zero. A utilização de dois elos para modelar essa operação serve para especificá-la com mais precisão. O elo que conecta os eventos de início de exibição possui ações do tipo *play sync* atribuídas aos seus pontos terminais destino para que em tempo de execução seja possível realizar a operação. No exemplo da Figura 3.18.c, o nó O2 foi definido para ser exibido simultaneamente com o nó O1.

É importante notar que o relacionamento *iniciar e terminar ao mesmo tempo* exige não apenas a compatibilização do posicionamento no tempo dos objetos, mas também a compatibilização da duração de suas exibições. No exemplo da Figura 3.18.c, a duração de exibição original do nó O2, foi alterada para que a operação de alto nível fosse satisfeita. Para os objetos que possuem tipo de mídia contínua (áudio e vídeo), e que sejam escolhidos para ser objetos dependentes de uma operação desse tipo, a compatibilização da duração de exibição pode acarretar, por exemplo, na inserção de intervalos de silêncio no caso de objetos de áudio, ou na repetição de quadros no caso de objetos de vídeo.

### 3.4.6 Browser de Base Privada

O browser de base privada é uma ferramenta do sistema HyperProp, que tem a função de permitir a edição gráfica e a navegação na estrutura de documentos hipermídia definidos de acordo com o Modelo de Contextos Aninhados [Much96]. O browser de base privada é um método da classe base privada do MCA. Como, geralmente, a rede de interconexões das bases privadas tende a ser razoavelmente complexa, deve-se utilizar um mecanismo para filtrar as informações necessárias para a visualização, no caso, a visão de olho-de-peixe [SaBr92] [Furn86].

Através do browser é possível criar novos nós, remover nós existentes, incluir e excluir componentes de nós de composição da base privada, e criar e remover elos definidos entre nós componentes da base privada em qualquer nível de aninhamento. A Figura 3.19 ilustra o exemplo da interface gráfica do browser de base privada do sistema HyperProp.

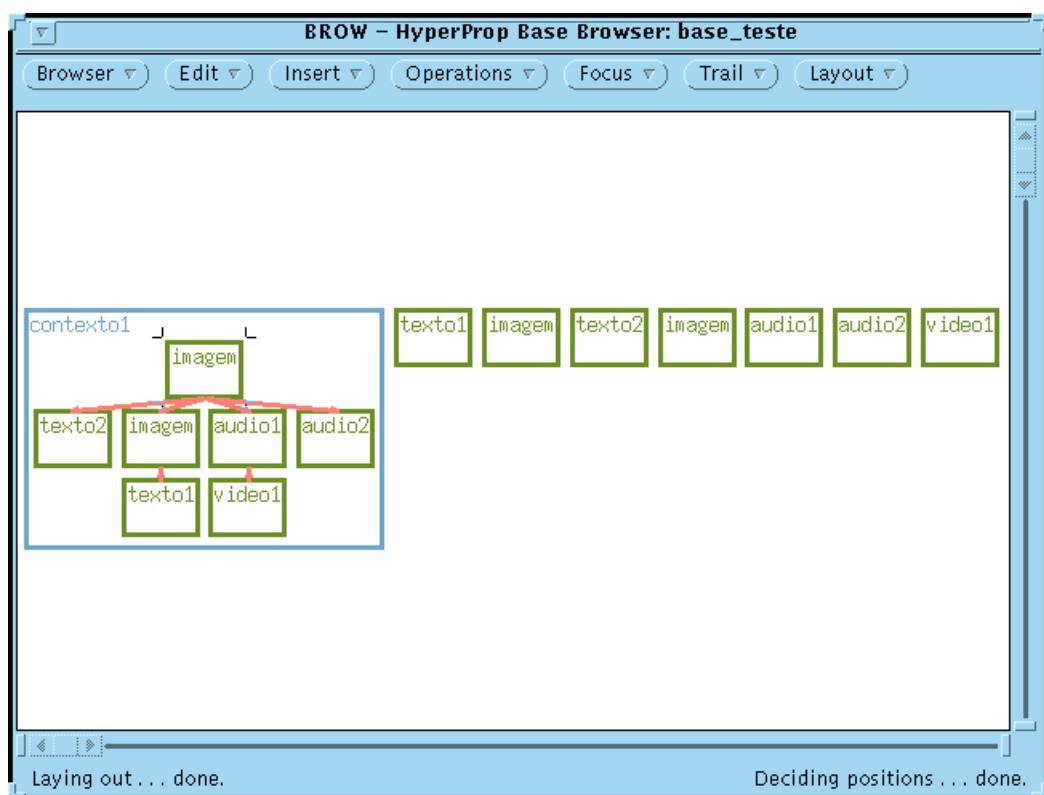


Figura 3.19: O browser de base privada do HyperProp

Durante o processo de navegação no browser de base privada, o usuário define qual será o nó em foco em um determinado instante. O conceito de nó em foco corresponde

ao nó acessado pelo usuário no qual ele deseja atribuir uma atenção especial. A partir da escolha do nó em foco, é realizada uma filtragem pelo browser entre os demais nós da base privada, calculando a função de grau de interesse que definirá quais demais nós constituirão a visão do browser. No exemplo da Figura 3.19 o nó em foco é o nó *imagem1*. Maiores detalhes sobre o mecanismo de filtragem e toda a concepção e implementação do browser de base privada podem ser obtidos em [Much96].

A definição do nó em foco é de grande relevância, pois é a partir da escolha do nó em foco pelo usuário no browser de base privada que, no EBS, as visões temporal e espacial irão se modificar. A interface entre o browser de base privada e o EBS será apresentada em detalhes no Capítulo 4.

# Capítulo 4

## O Editor e Browser de Sincronismo

### 4.1 Introdução

A motivação principal desta dissertação é construir uma ferramenta gráfica, o Editor e Browser de Sincronismo (EBS), que auxilie o usuário na definição das sincronizações temporal e espacial de documentos hipermídia definidos através do Modelo de Contextos Aninhados.

O EBS é uma ferramenta que faz parte de um ambiente para definição de sincronismo e controle de apresentação de hiperdocumentos, chamado *HySEE (HyperProp Show Editor and Executor)* [SoSo96], componente do sistema hipermídia *HyperProp* [Soar95].

O ambiente de edição e execução de apresentações multimídia *HySEE* é composto pelos módulos de edição, exibição e execução, como mostra a Figura 4.1.

O autor tem a seu dispor, no módulo de edição, duas formas para descrever suas apresentações: textualmente, através das linguagens SOD (Show Object Description) e SDD (Show Dynamic Description), ou graficamente, através da linguagem SGD (Show Graphical Description) do EBS [SoSo96].

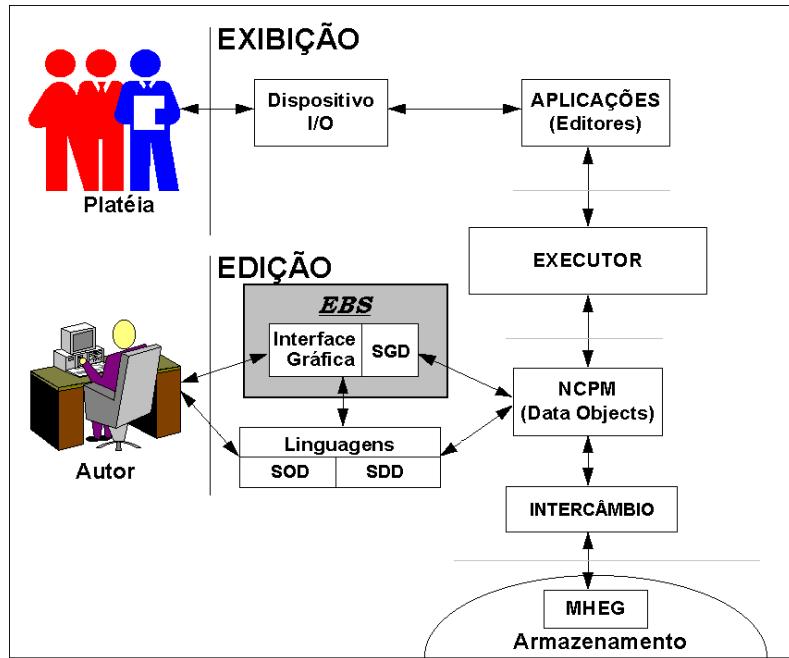


Figura 4.1: Uma visão geral do ambiente *HySEE* e do EBS

A descrição da apresentação nas linguagens SOD, SDD ou SGD é traduzida para entidades MCA, que se constituem na representação interna do ambiente *HySEE* de uma apresentação multimídia. O módulo executor é o responsável pelo controle da execução das apresentações representadas em objetos MCA. No módulo de intercâmbio, a representação MCA dos documentos hipermídia é traduzida para a representação MHEG. A contextualização do sistema EBS dentro do ambiente *HySEE* pode ser visualizada pela região destacada na Figura 4.1.

A idéia que motivou o desenvolvimento do EBS foi tornar mais fácil para o usuário a tarefa de editar a apresentação de um documento hipermídia. No EBS, toda a especificação feita pelo usuário através da linguagem SGD passa por um processo de validação. O sistema reporta inconsistências e adverte para situações onde possa haver erros, à medida que a especificação vai sendo traduzida para entidades MCA. Em outras palavras, o EBS realiza uma compilação incremental durante o processo de edição de um documento.

Neste capítulo todas as funções do EBS são abordadas, e as características da ferramenta para a definição da sincronização temporal e espacial serão detalhadas. Na Seção 4.2 é mostrada a interface gráfica do EBS. Na Seção 4.3 é apresentado o

ambiente integrado para edição e navegação do sistema HyperProp composto pelo EBS e pelo browser de base privada.

## 4.2 A Interface Gráfica do EBS

No ambiente do EBS, o autor pode editar a apresentação de um documento através de uma interface gráfica, que fornece mecanismos para o posicionamento dos componentes do documento no tempo e no espaço. A interface gráfica do EBS é composta por duas janelas:

- *Time View*;
- *Spatial View*.

A janela *Time View* do EBS tem por finalidade oferecer mecanismos especiais, por meio de representações gráficas, para a definição da sincronização temporal relativa entre os componentes de um hiperdocumento baseada nas entidades do subsistema de apresentação do HyperProp. A janela Time View será o foco da Subseção 4.2.1, onde suas funcionalidades serão apresentadas.

A janela *Spatial View* do EBS, que será especificada em detalhes na Subseção 4.2.2, oferece mecanismos para a definição da sincronização espacial dos componentes de um hiperdocumento baseada nas entidades do subsistema de apresentação do HyperProp. As duas janelas de interface do EBS, a Time View e a Spatial View, funcionam de forma integrada, e essa integração será apresentada no decorrer da Subseção 4.2.2 e na Seção 4.3.

### 4.2.1 A Time View

A Time View é a janela de interface gráfica do EBS onde o usuário pode definir os relacionamentos temporais relativos entre os componentes de um hiperdocumento, utilizando as entidades do subsistema de apresentação do HyperProp.

Na Time View os componentes de um hiperdocumento são colocados no eixo do tempo a cada interação feita pelo usuário. É importante ressaltar que todos os relacionamentos temporais, realizados na janela Time View do EBS, interconectam os

objetos entre si, e não em relação ao próprio eixo do tempo, fazendo com que o posicionamento temporal de cada objeto seja sempre relativo a outro objeto. A Figura 4.2 mostra o layout utilizado na janela Time View.

No layout da Time View, o eixo do tempo é dividido em tempo positivo e tempo negativo, para que seja possível visualizar todos os componentes de uma determinada visão temporal de um hiperdocumento, conforme será discutido na Subseção 4.2.1.1.

A *Time View* possui ainda outra região, chamada *região de limbo*, que pode ser visualizada na Figura 4.2. Nessa região podem ser colocados nós cujos momentos exatos dos inícios de suas exibições são impossíveis de precisar, conforme será visto mais adiante.

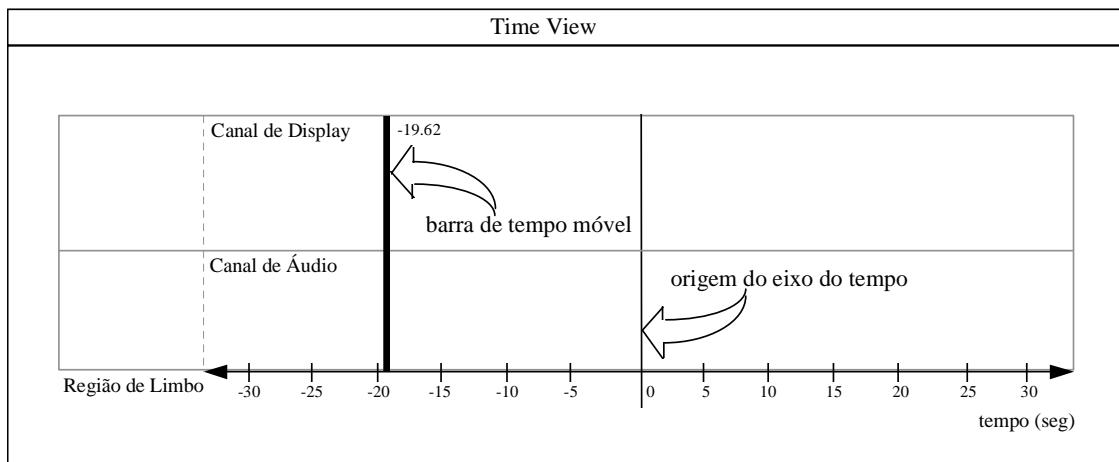


Figura 4.2: Layout da janela Time View do EBS

A *Time View* possui uma barra de tempo móvel que pode ser posicionada em qualquer instante do tempo. Na Figura 4.2, a barra de tempo encontra-se no instante -19.62 segundos, conforme indicado pelos números posicionados no canto superior da barra. Se a barra de tempo estiver sobre um ou mais nós, estes aparecerão na janela *Spatial View* nos canais em que eles deverão ser exibidos, conforme será detalhado na Subseção 4.2.2.

Os nós são representados por retângulos, cujo comprimento indica a duração esperada no tempo para sua exibição. Os elos são representados por arestas que conectam os nós. Os objetos ficam dispostos nas regiões *canal de áudio* e *canal de display*,

representadas por faixas mostradas na Figura 4.2, que correspondem às abstrações dos dispositivos de saída de vídeo e áudio da plataforma de exibição.

Quando um nó é selecionado (*mousedown-mouseup*) na estrutura de um documento (através do browser de base privada), é criada, pelo EBS, uma visão do sincronismo na *Time View*, focada nesse *nó*, denominado *objeto base*. O *objeto base* é posicionado na origem do eixo do tempo, conforme pode ser visto no exemplo da Figura 4.3 pela seleção do nó O1.

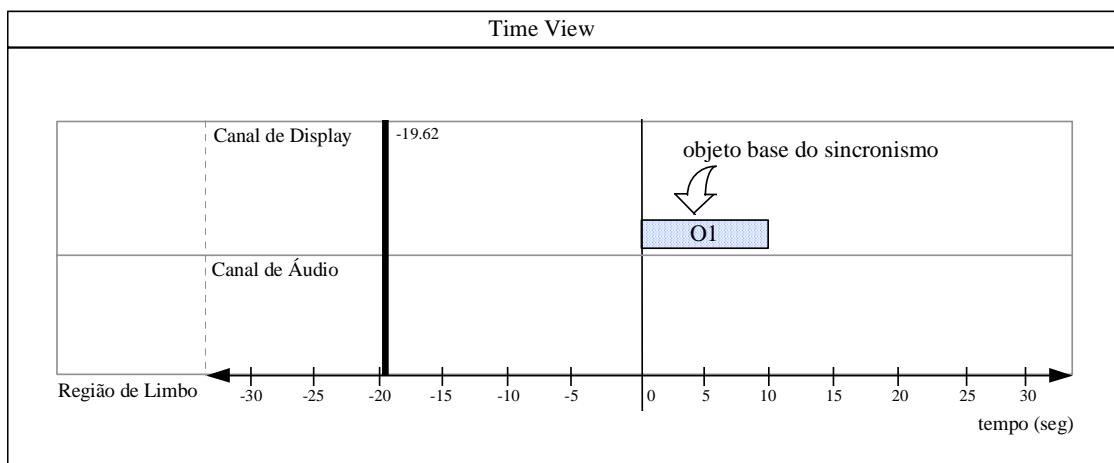


Figura 4.3: O objeto base da Time View

Todos os demais nós que serão incluídos na Time View estão, necessariamente, relacionados com o objeto base por elos de sincronização. As regras para a inclusão de nós na Time View serão analisadas na próxima subseção.

As seguintes operações da Time View serão analisadas nas próximas subseções:

- inclusão de objetos na Time View;
- criação de elos de sincronização;
- criação de relações de alto nível;
- manutenção da consistência temporal;

#### 4.2.1.1 Inclusão de Objetos na Time View

Os nós apresentados na Time View são instâncias dos objetos de dados correspondendo aos objetos de representação. Cada nó incluído na Time View corresponde à uma instância diferente de um mesmo nó (objeto de dados), ou de nós diferentes. Note que, segundo a Figura 3.10, o EBS trabalha no plano de objetos de representação.

Os relacionamentos definidos através de elos de hipermídia não são mostrados na Time View, pois a condição de ocorrência desse tipo de elo é assíncrona, como por exemplo, uma interação do usuário. A janela Time View do EBS mostra apenas os relacionamentos temporais modelados por elos de sincronização, nos quais é possível precisar com exatidão o momento em que os eventos envolvidos nos elos ocorrem.

Quando o objeto base é selecionado, ou quando um nó qualquer é incluído na Time View, o EBS pode incluir automaticamente os demais nós (posteriores e anteriores) que estão conectados a esse nó através de elos de sincronização, desde que as regras de inclusão de nós sejam satisfeitas. Tais regras são apresentadas abaixo:

- Inclusão de nós antecessores: Se um nó incluído na visão for destino de um elo de sincronização 1:n, o nó origem desse elo é incluído na visão na posição determinada pelo elo.
- Inclusão de nós posteriores: Se um nó é incluído na visão, todos os nós de destino de seus elos de sincronização cujas condições de disparo estão satisfeitas, são também incluídos na visão.

Note que essas regras de inclusão são recursivas e são aplicadas a todos os nós que são incluídos na visão do sincronismo da *Time View*, iniciando-se pelo objeto base.

Um exemplo da inclusão recursiva de nós na Time View pode ser visualizado na Figura 4.4. Suponha que o usuário selecionou o nó O3 para ser o objeto base do sincronismo. Os nós O2, O4 e O5 também são incluídos na visão por satisfazerem as regras de inclusão apresentadas anteriormente. Entretanto, os nós O1, O6 e O7 não são incluídos na Time View, pois estão conectados aos nós O2, O4 e O5 através de elos de hipermídia, não sendo possível precisar os instantes de suas exibições.

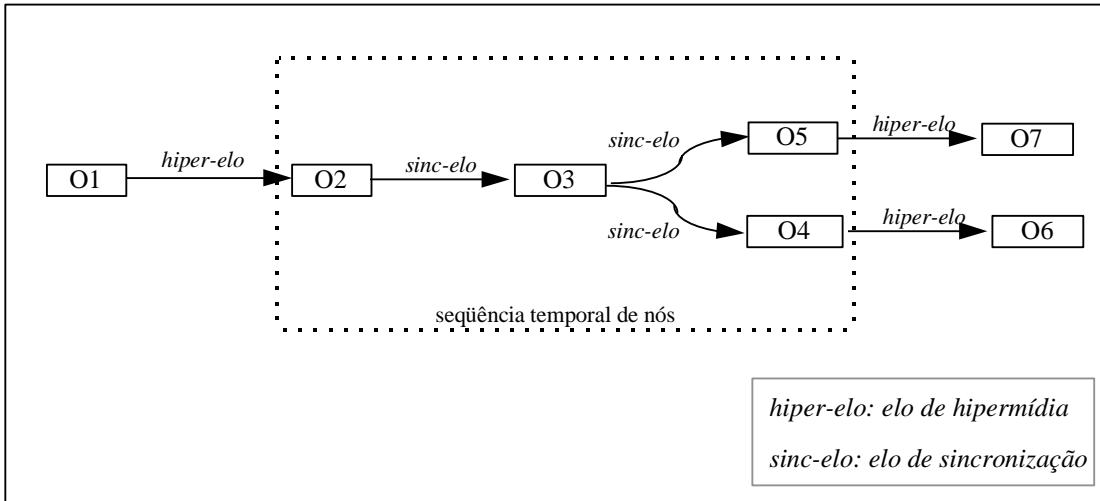


Figura 4.4: Um exemplo de uma seqüência temporal de nós

Os nós que são mostrados pela Time View formam uma seqüência temporal de nós. No caso da Figura 4.4, os nós O2, O3, O4 e O5 constituem a seqüência temporal de nós da Time View.

No exemplo da Figura 4.5, quando o nó O1 foi escolhido para ser o objeto base, os nós O2, O3 e O4, formadores da seqüência temporal de nós, também são incluídos automaticamente na Time View pela aplicação das regras citadas anteriormente. Nesse exemplo, os elos *elo1*, *elo2*, e *elo3* são elos de sincronização.

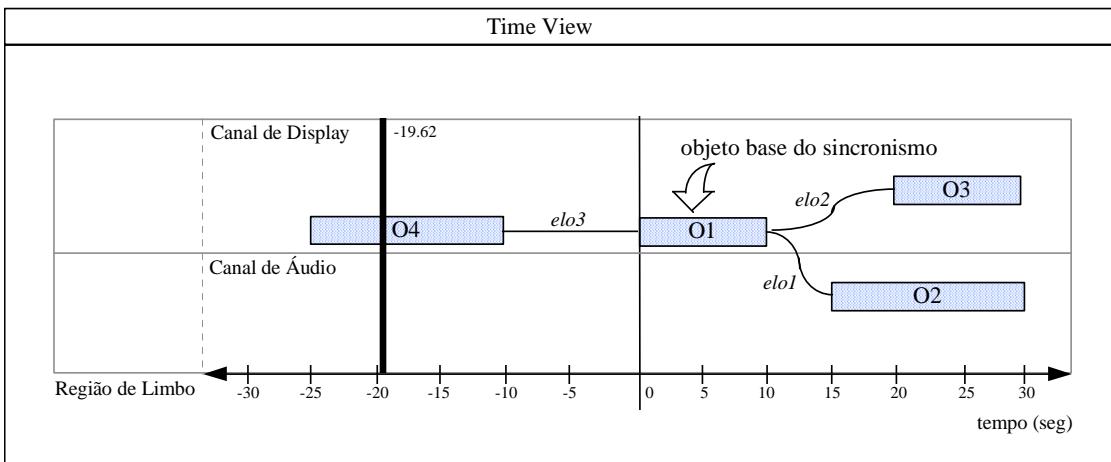


Figura 4.5: Inclusão automática de nós na Time View

O usuário também pode incluir explicitamente um nó situado no browser de base privada (repositório de dados) dentro da visão atual da Time View através da operação de arrastamento (*mousedown-drag-mouseup*). Quando um nó *N* é incluído na Time

View, ele é posicionado inicialmente na origem do eixo do tempo. O EBS cria automaticamente um elo de sincronização (relação 1:1) tendo o *objeto base* como a extremidade de origem e  $N$  como a extremidade de destino.

A Figura 4.6 ilustra um exemplo onde o nó O5 foi incluído explicitamente pelo usuário na Time View, posicionando-o na origem do eixo do tempo. O elo *elo4* (relação 1:1) foi criado automaticamente pelo EBS, conectando o objeto base, nó O1, ao nó incluído, o nó O5.

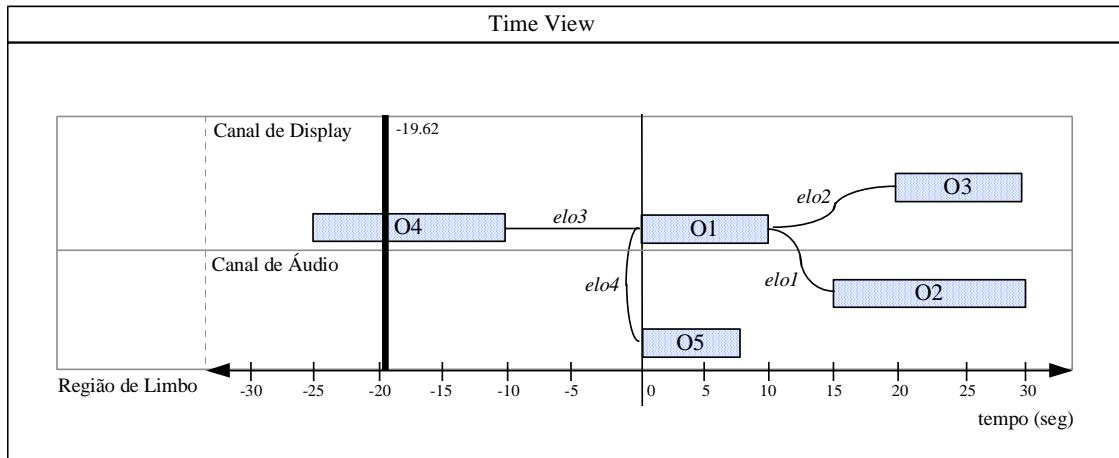


Figura 4.6: Inclusão explícita de um nó na Time View

Os nós também podem ser incluídos na região de limbo. Nessa região são posicionados os nós cujos eventos não se pode precisar a priori. A colocação desses nós é realizada através de operações de arrastamento, como mencionadas anteriormente. Quando um nó é arrastado para o limbo, não é criado nenhum elo entre ele e o objeto base. Todos os eventos contidos nos objetos pertencentes ao *limbo* são considerados como tendo ocorrido em algum momento passado, por definição, embora não se possa definir quando. Objetos no limbo vão assim permitir que relações de sincronismo possam ser editadas ou visualizadas, mesmo que não se saiba o exato momento de ocorrência de alguns de seus eventos de origem.

No exemplo da Figura 4.7, o nó O4 foi posicionado na região de limbo para que, posteriormente, possa ser relacionado com algum outro nó pertencente a seqüência temporal de nós da Time View, através da seleção de opções de menu, como será visto na subseção seguinte.

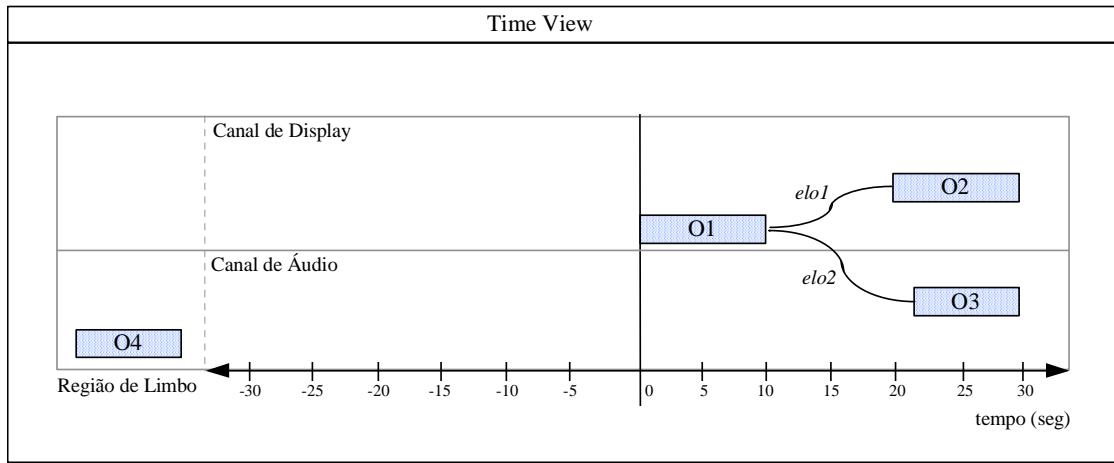


Figura 4.7: Inclusão de um nó na região de limbo

#### 4.2.1.2 Criação de Elos de Sincronização

Outra facilidade oferecida pela Time View, é a criação de elos de sincronização para interconectar objetos de representação contidos na visão temporal. Um elo pode ser criado, relacionando nós localizados na Time View. Quando um elo de sincronização é criado, os nós são posicionados no tempo de acordo com os instantes especificados pelo elo.

Na Figura 4.8, é ilustrada a criação do elo *elo5* (relação 1:1), que conecta o nó O1 ao nó O5. A criação do elo *elo5* acarretou a destruição automática do elo *elo4*, conforme será explicado na Subseção 4.2.1.4.

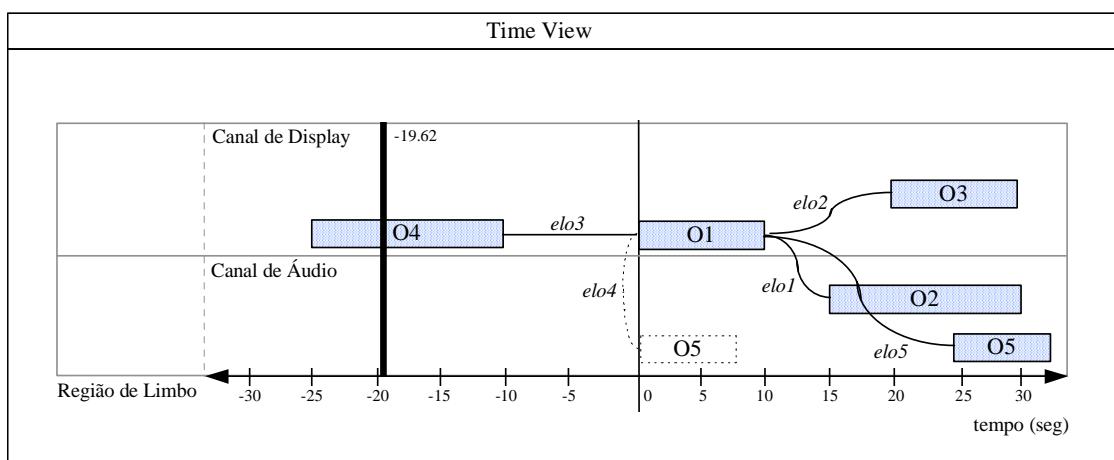


Figura 4.8: Criação de um elo de sincronização

#### 4.2.1.3 Criação de Relações de Alto Nível

Dentro da janela Time View o usuário pode criar relações temporais de alto nível entre os nós utilizando as opções de menu. Essas relações correspondem aos operadores descritos anteriormente na Subseção 3.4.5, *exibir ao término de*, *exibir iniciando ao mesmo tempo* e *exibir iniciando e terminando ao mesmo tempo*.

Como já foi explicado, a criação de relações de alto nível implica na criação automática de elos de sincronização entre os nós envolvidos na relação. Os atributos desses elos são definidos automaticamente pelo EBS (ver Subseção 3.4.5).

Para exemplificar como o usuário deve utilizar a interface gráfica da Time View para criar as relações de alto nível, serão ilustrados a seguir exemplos para cada uma das relações.

O exemplo da Figura 4.9 supõe que o usuário deseja criar uma relação do tipo *exibir ao término de* entre o nó O2 e o nó O3, que foi incluído inicialmente na origem do eixo do tempo (linhas pontilhadas na figura). O usuário utilizará o menu para criar tal relação (janela *popup* que aparece na figura), especificando qual será o objeto base da relação e qual(ais) será(ão) o(s) objeto(s) dependente(s). No exemplo da Figura 4.9 foi criada a relação(1:1) do tipo: *exibir O3 ao término de O2*. O EBS posicionou automaticamente o início da exibição do nó O3 no fim da exibição do nó O2, e criou o elo *elo3*, que conecta o evento de fim de exibição do nó O2 com o início da exibição do nó O3.

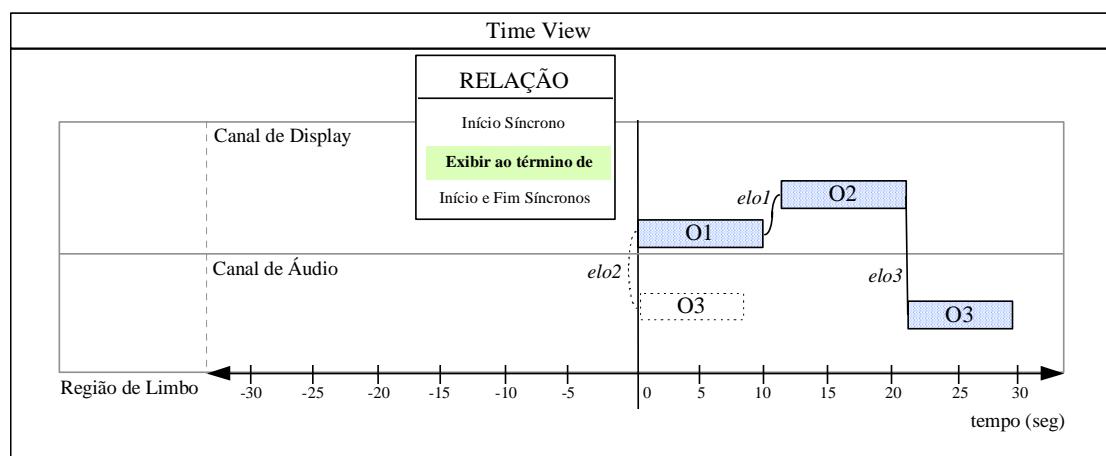


Figura 4.9: Criação da relação *exibir ao término de*

O exemplo da Figura 4.10 supõe que o usuário deseja criar uma relação do tipo *exibir iniciando ao mesmo tempo* entre o nó O2 e o nó O3, que foi incluído inicialmente na origem do eixo do tempo (linhas pontilhadas na figura). O usuário também utilizará o menu para criar tal relação (janela *popup* que aparece na figura). No exemplo da Figura 4.10 foi criada a relação(1:1) do tipo: *exibir O3 iniciando ao mesmo tempo de O2*. O EBS posicionou automaticamente o início da exibição do nó O3 com o início da exibição do nó O2, e criou o elo *elo3*, que conecta o evento de início da exibição do nó O2 com o início da exibição do nó O3.

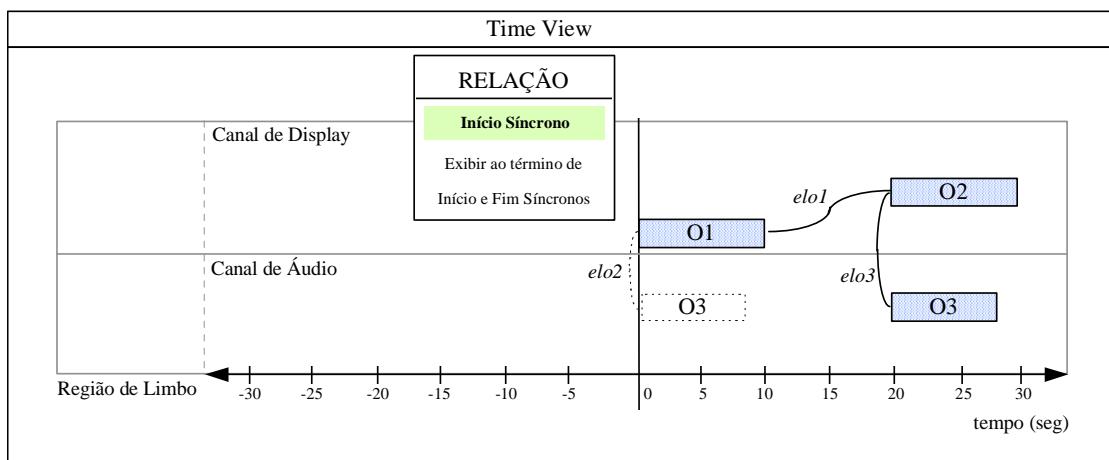


Figura 4.10: Criação da relação exibir iniciando ao mesmo tempo

O exemplo da Figura 4.11 supõe que o usuário deseja criar uma relação do tipo *exibir iniciando e terminando ao mesmo tempo* entre o nó O2 e o nó O3, que foi incluído inicialmente na origem do eixo do tempo (linhas pontilhadas na figura). O usuário também utilizará as opções de menu para criar tal relação. No exemplo da Figura 4.11 foi criada a relação(1:1) do tipo: *exibir O3 iniciando e terminando ao mesmo tempo de O2*. O EBS posicionou automaticamente o início da exibição do nó O3 com o início da exibição do nó O2 e também o fim da exibição do nó O3 com o fim da exibição do nó O2. O EBS criou os elos *elo3* e *elo4*, que conectam, respectivamente os eventos de início e fim da exibição dos nós O2 e O3.

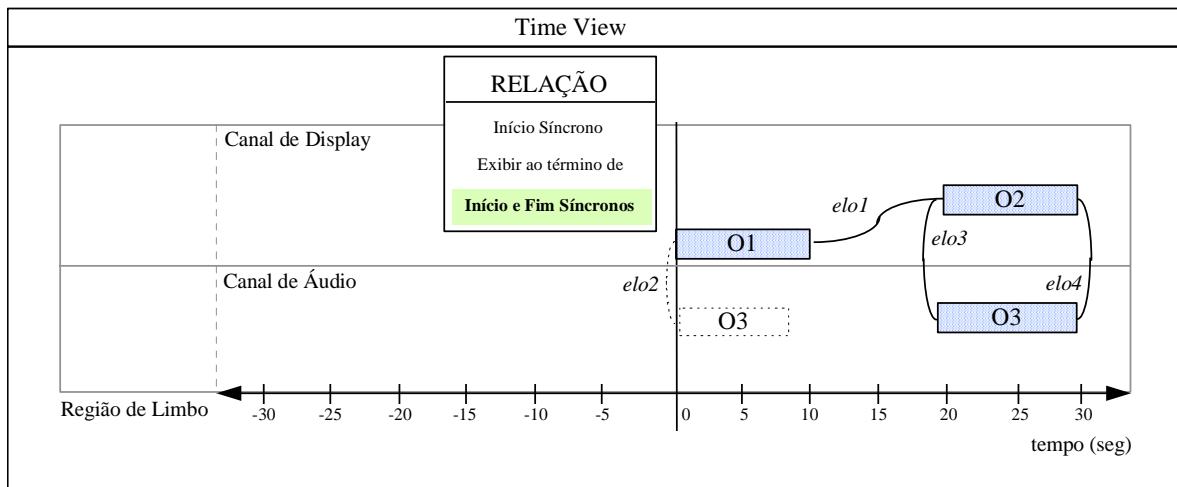


Figura 4.11: Criação da relação exibir iniciando e terminando ao mesmo tempo

Note que o relacionamento *iniciar e terminar ao mesmo tempo* exige não apenas a compatibilização do posicionamento do tempo dos objetos, mas também a compatibilização da duração de suas exibições. Veja que, no exemplo da Figura 4.11, tornou-se necessário aumentar a duração da exibição do nó O3. Dependendo do tipo de mídia apresentada pelo nó O3, pode não ser possível realizar tal operação. Se o nó O3 corresponder, por exemplo, a exibição de uma imagem gráfica, aumentar o tempo de sua duração é uma tarefa fácil de ser realizada, pois trata-se de uma mídia estática, porém, se o nó O3 corresponder a exibição de um vídeo (mídia contínua), aumentar sua duração de exibição pode não ser possível. Uma alternativa, nesse caso, seria repetir a exibição do nó O3.

O EBS pede a confirmação do usuário, por meio de uma janela gráfica, antes de processar qualquer relação de alto nível.

#### 4.2.1.4 Mantendo a Consistência Temporal de um Hiperdocumento

O processo de definição da sincronização temporal deve manter a consistência da especificação da apresentação do hiperdocumento feita pelo autor após qualquer interação com a interface gráfica da janela Time View. O EBS reposiciona todos os objetos na Time View e reporta possíveis inconsistências na definição da sincronização temporal.

O usuário pode alterar o posicionamento relativo dos objetos, desde que seja mantida a consistência do sincronismo. A consistência da definição de sincronismo no EBS não permite, por exemplo, que um evento de um objeto destino de um elo seja posicionado antes do evento origem desse mesmo elo.

Existem várias operações de interface gráfica que o usuário pode realizar na Time View. O processo de manutenção da sincronização temporal é ativado para todas as operações. Dentre elas, destacam-se:

- alteração da duração da exibição de um nó;
- alteração no posicionamento temporal de um nó;
- criação de um elo de sincronização;
- remoção de um elo de sincronização;
- alteração de um elo de sincronização.

A Figura 4.12 ilustra o processo de manutenção da consistência temporal na janela Time View após a alteração na duração da exibição do nó O1. As linhas pontilhadas mostram a situação antes da mudança da duração. Note que o nó O1 possuía, antes da alteração na duração, aproximadamente 7 segundos de duração, e após a mudança, passou para aproximadamente 13 segundos. Note que todos os nós dependentes temporalmente do nó O1 (O2, O3 e O5) são repositionados automaticamente.

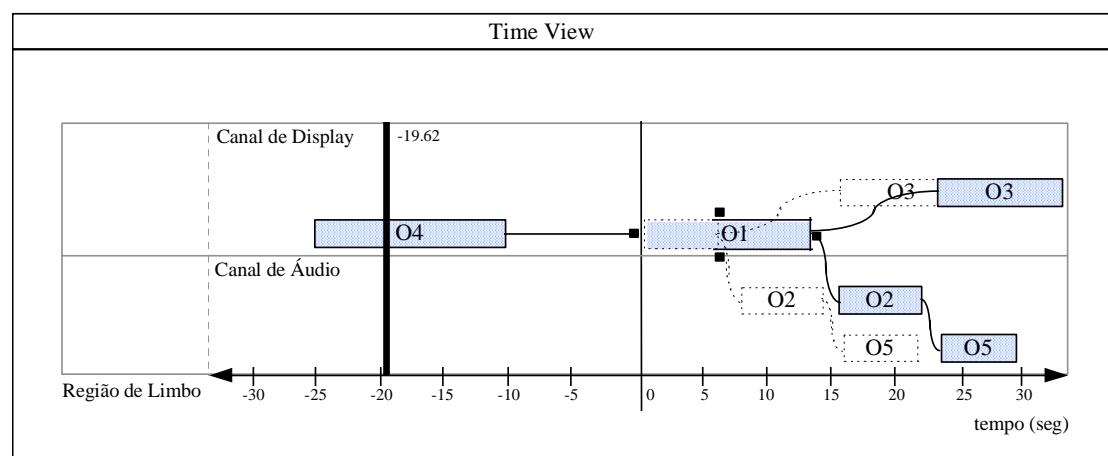


Figura 4.12: Alteração na duração do nó O1

A Figura 4.13 mostra outro exemplo onde é necessário manter a consistência inicial do relacionamento temporal entre os objetos. Nessa figura o usuário mudou o posicionamento temporal apenas do objeto  $O2$ . Neste caso, o EBS recalcula o elo  $elo1$  que é o elo adjacente ao objeto  $O2$ , de maneira a manter o posicionamento inicial do objeto  $O2$  indicado pelo elo.

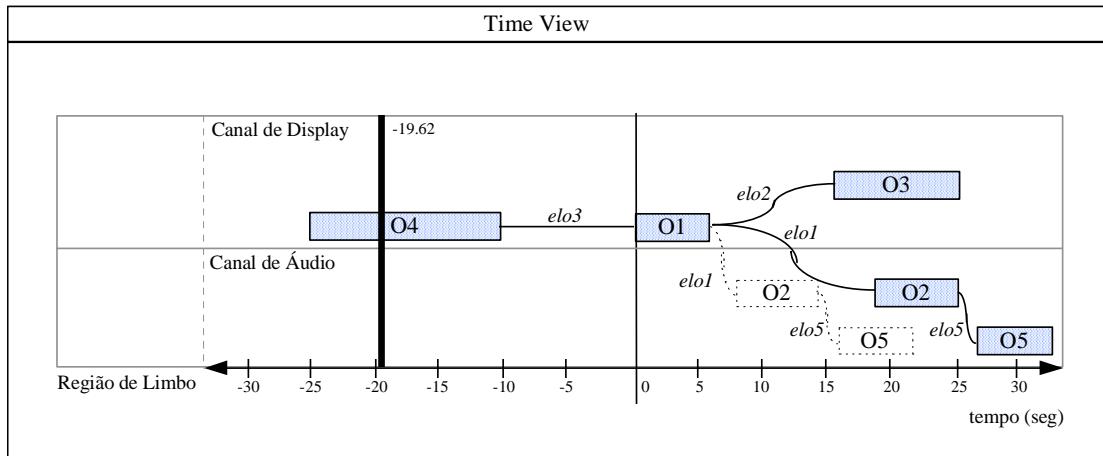


Figura 4.13: Repositionamento temporal do nó  $O2$

Na determinação da consistência, o elo de sincronização criado entre o objeto base e um nó  $N$  arrastado tem tratamento diferenciado. Se o nó  $N$  arrastado do *browser de base privada* tiver seu posicionamento alterado pela criação de um elo de sincronização do qual é destino, o elo de sincronização com o objeto base é simplesmente destruído, passando sua posição no tempo a ser definida pelo elo recém criado.

Na Figura 4.14, o posicionamento do nó  $O3$  é alterado pela criação do elo  $elo_3$  que conecta o nó  $O1$  ao nó  $O3$ . Neste caso, o EBS destrói automaticamente o elo  $elo_1$  criado automaticamente do nó  $O1$ , objeto base da visão, para o objeto  $O3$ , quando este foi trazido para a *Time View*.

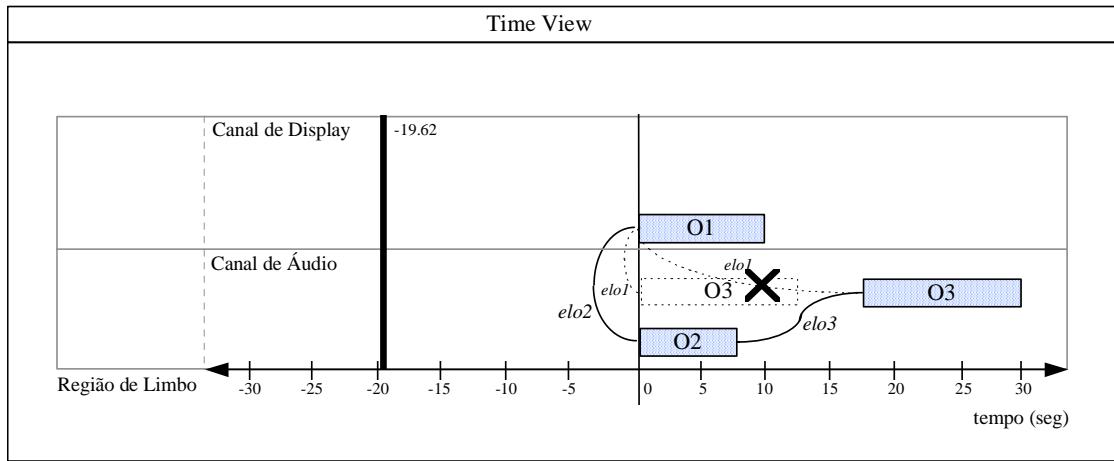


Figura 4.14: A criação do elo *elo3* implica na destruição do elo *elo1*

Quando o usuário destrói um elo  $E$  na *Time View*, os nós e elos posicionados a partir de  $E$  podem deixar de ser mostrados, pois, neste caso, a seqüência temporal de nós da *Time View* original que começava no objeto base pode deixar de existir como consequência da exclusão de  $E$ .

No exemplo da Figura 4.15, o elo *elo1* foi destruído pelo usuário e, como consequência, a seqüência temporal de nós formada pelos nós  $O2$  e  $O5$  e pelo elo *elo5* é retirada da visão, pois não mais pertence à seqüência temporal de nós da *Time View*.

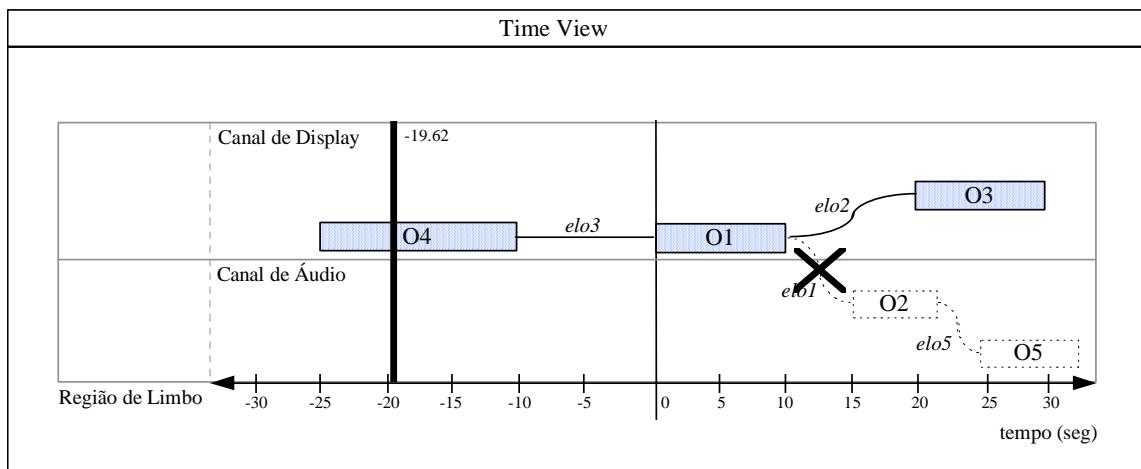


Figura 4.15: A remoção do elo *elo1*

#### 4.2.2 A Spatial View

Na *Spatial View*, o autor edita o sincronismo espacial, posicionando os objetos dentro dos canais onde eles serão apresentados. É através deste posicionamento que o autor

define, por exemplo, o espaço do display que será usado para exibir uma janela que contém uma imagem gráfica, ou o volume que será usado para exibir um áudio, no instante de tempo definido pela barra de tempo da *Time View*.

Para definir o espaço em um dispositivo que será usado para exibir um objeto, o usuário deve posicionar o retângulo que o representa na janela (canal) associada ao dispositivo. O significado desse posicionamento varia conforme o tipo do canal. Existem duas janelas de interface na *Spatial View*:

- janela *Canal de Display*, onde são posicionados os nós que serão exibidos no monitor de vídeo;
- janela *Canal de Áudio*, onde são indicados os percentuais de exibição dos nós que serão exibidos no dispositivo de áudio.

O layout da janela *Canal de Display* é mostrado na Figura 4.16. A resolução suportada pela janela é indicada na parte superior (no exemplo a resolução é de 800x600 pixels). Para os objetos que serão exibidos em um monitor de vídeo — vídeos, imagens gráficas e textos — o retângulo que representa um objeto na *Spatial View* indica a localização da janela onde o objeto será exibido no monitor. Os nós, representados na figura por regiões hachuradas (janelas), são posicionados pelo usuário dentro da janela *Canal de Display* e podem ser redimensionados e repositionados livremente.

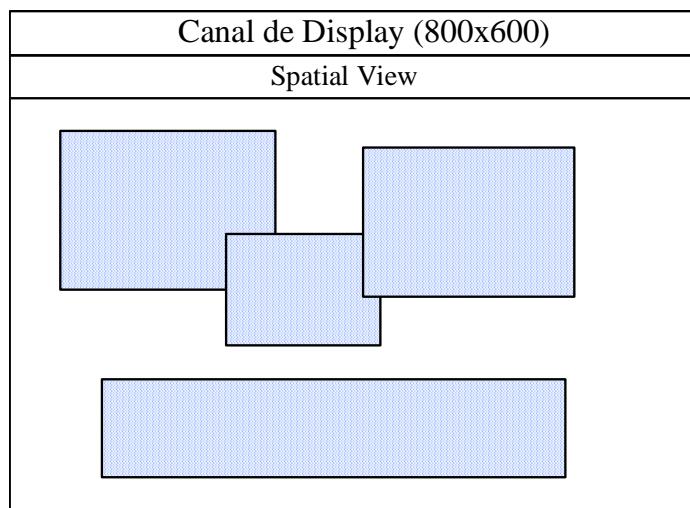


Figura 4.16: Layout da janela *Canal de Display*

O layout da janela *Canal de Áudio* é mostrado na Figura 4.17. Os retângulos hachurados indicam os percentuais do volume de exibição máximo usado na reprodução dos conteúdos dos objetos representados no dispositivo de áudio. No caso de dois ou mais objetos de áudio exibidos em um mesmo canal ao mesmo tempo, o tamanho dos retângulos indica o percentual utilizado na mixagem dos sinais na reprodução.

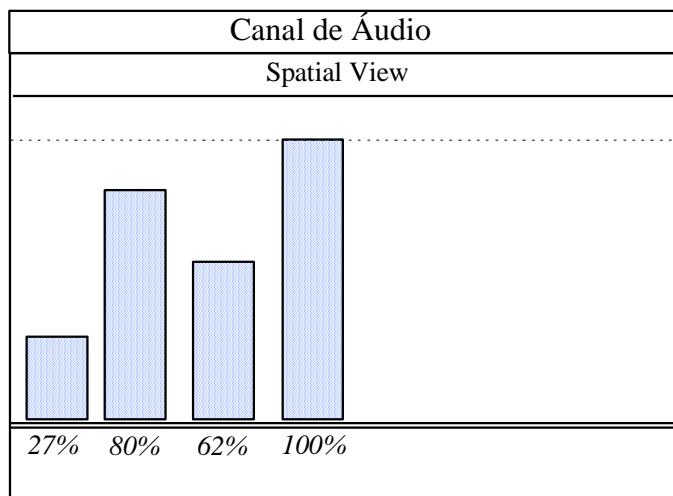


Figura 4.17: Layout da janela *Canal de Áudio*

As visões das janelas Time View e Spatial View funcionam de forma coordenada. A barra de tempo da janela Time View indica a disposição de momento na Spatial View, mostrada nas janelas canal de áudio e canal de display.

No exemplo da Figura 4.18, a barra de tempo da janela Time View encontra-se posicionada sobre os nós O3, O4, O5 e O6, que são mostrados nas janelas canal de display (O3 e O4), e canal de áudio (O5 e O6).

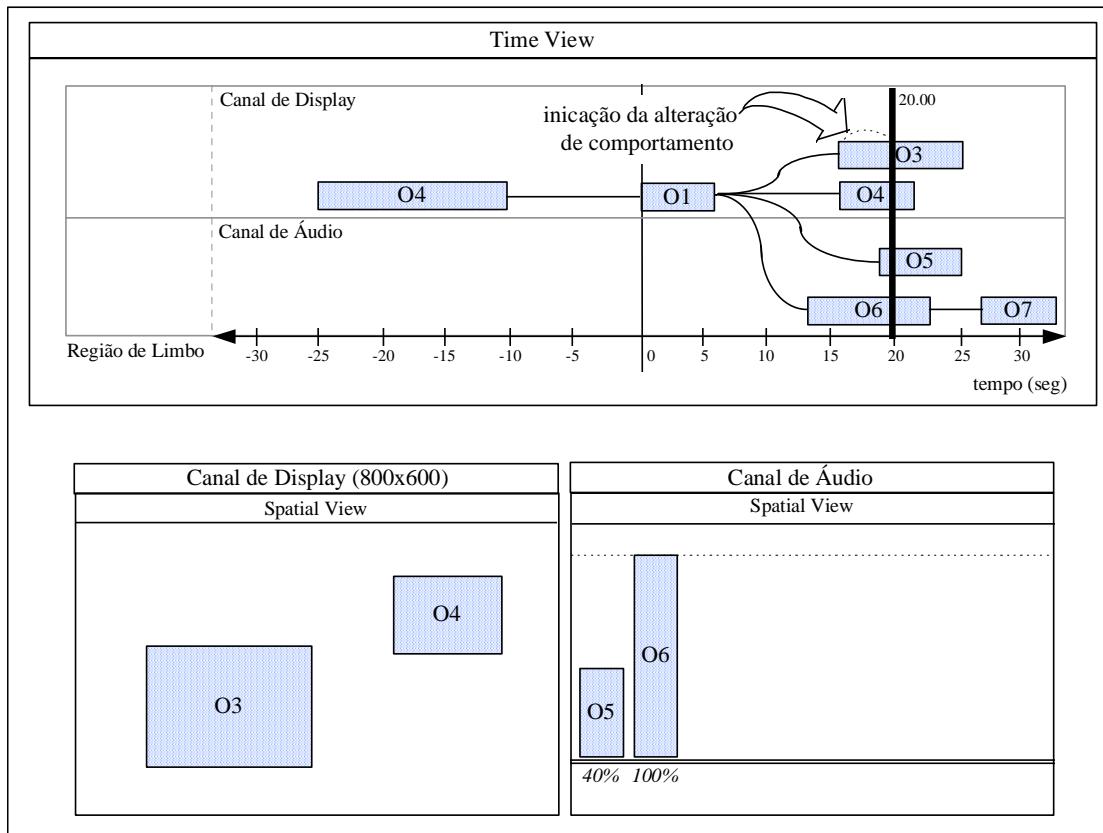


Figura 4.18: Relacionamento entre a Time View e a Spatial View

Quando o usuário move a barra de tempo na Time View, a disposição dos objetos na Spatial View também é modificada de acordo com as mudanças de comportamento espacial definidas. O funcionamento coordenado das visões temporal e espacial de um hiperdocumento facilita a visualização da apresentação feita pelo usuário.

As seguintes operações da Spatial View serão analisadas nas próximas subseções:

- reposicionamento espacial de objetos;
- redimensionamento espacial de objetos;
- alteração de volume de exibição para objetos de áudio;

#### 4.2.2.1 Display Channel

No canal de display, podem ser definidas alterações do comportamento espacial dos nós que serão exibidos no monitor de vídeo — vídeos, imagens gráficas e textos. Os

tipos de alteração de comportamento definidas pelo usuário dentro da janela canal de display da Spatial View são:

- reposicionamento das janelas que representam os nós a serem exibidos;
- redimensionamento das janelas que representam os nós a serem exibidos.

É importante ressaltar que o EBS cria eventos de alteração de comportamento para todas as interações que o usuário realiza na Spatial View, podendo um evento de alteração de comportamento ser associado a qualquer região do nó. Todos os eventos de alteração de comportamento definidos na Spatial View são inseridos na lista de eventos dos descritores dos nós de representação em questão.

No exemplo mostrado na Figura 4.18, o usuário definiu uma alteração de comportamento espacial do nó O3 no momento indicado pela barra de tempo (20 segundos). Essa alteração de comportamento foi associada ao evento de início de exibição do nó O3 (região  $\lambda$ ), e é representada na Time View por uma linha pontilhada (destacada na Figura 4.18).

A Figura 4.19 ilustra uma alteração de comportamento efetuada pelo usuário. Nesse exemplo, o posicionamento do nó O1 foi alterado através de uma operação de arrastamento (*mouseon-mouseoff*).

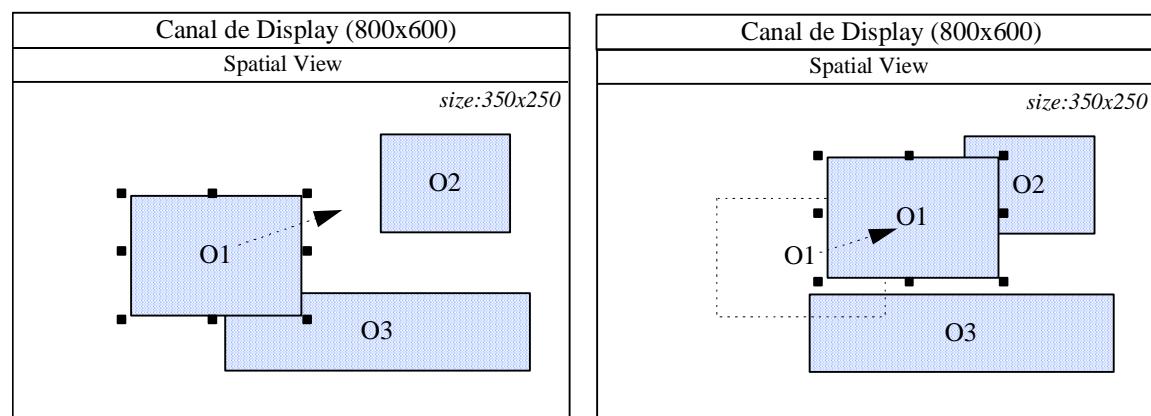


Figura 4.19: Repositionamento espacial de objetos na Spatial View

O exemplo da Figura 4.20 mostra a operação de redimensionamento espacial de objetos no canal de display da Spatial View. Essa operação pode ser efetuada através

do redimensionamento da janela que representa o nó “esticando” as bordas do objeto com o mouse.

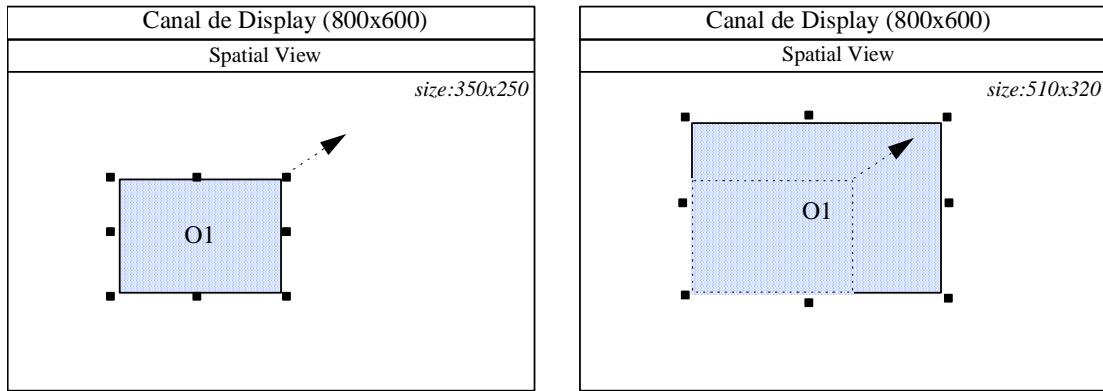


Figura 4.20: Redimensionamento espacial de objetos na Spatial View

#### 4.2.2.2 Audio Channel

No canal de áudio, também são definidas alterações do comportamento de exibição espacial dos nós de áudio. Apenas um tipo de operação pode ser definida no canal de áudio: a alteração do volume de exibição de um nó.

A Figura 4.21 ilustra uma alteração de comportamento efetuada pelo usuário. Nesse exemplo, o volume de exibição do nó O2 foi alterado de 100% de exibição para 50% através de uma operação de “encurtamento” na borda do nó com o mouse. É importante ressaltar que o percentual indicado no canal de áudio não se refere ao volume total do dispositivo de exibição, mas sim o percentual em cima do volume de exibição do próprio nó.

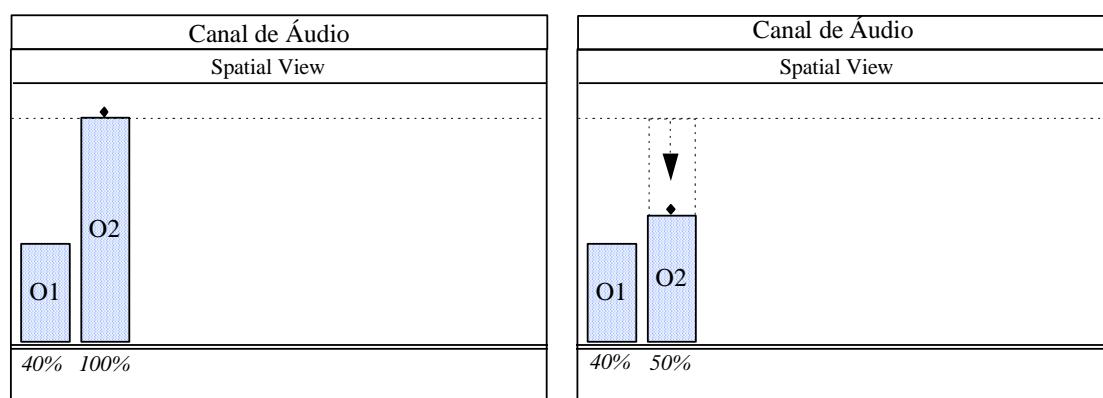


Figura 4.21: Alteração do volume de exibição de objetos de áudio na Spatial View

### 4.3 Ambiente Integrado de Edição do HyperProp

A edição de documentos hipermídia pode ser analisada sob três visões distintas. A primeira visão considera a edição da estrutura de um hiperdocumento, fornecendo recursos para editar nós e elos e para agrupá-los em composições, caso o modelo conceitual hipermídia permita composições aninhadas. A segunda visão é responsável por especificar as relações temporais entre os componentes de um hiperdocumento, definindo o posicionamento relativo dos componentes no tempo. E a terceira visão define as relações espaciais dos componentes de um hiperdocumento, estabelecendo suas características de exibição em um determinado dispositivo.

Para oferecer ao usuário um ambiente completo para manipulação de hiperdocumentos deve-se possibilitar a definição e visualização de suas relações estruturais e também de suas relações de sincronização temporais e espaciais. No sistema HyperProp, isso é construído integrando-se o browser de base privada (browser de estrutura) e o EBS (browser de sincronismo), como apresentado na Figura 4.22.

As duas ferramentas apresentam visões de um hiperdocumento baseadas em um determinado nó que representa o interesse atual do usuário em um momento do processo de autoria ou navegação. No browser da estrutura, a visão exibida é computada de acordo com o nó em foco pelo usuário, sendo fornecidas informações globais sobre o hiperdocumento e detalhes locais referentes a este nó. No browser de sincronismo, a visão apresentada é baseada neste mesmo nó e é composta por todos os demais objetos a ele relacionados no tempo.

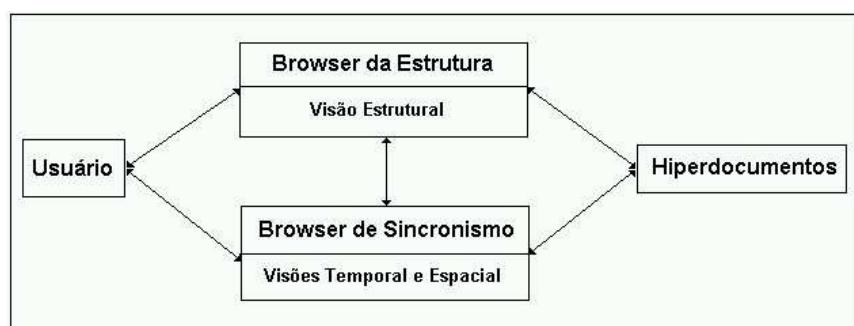


Figura 4.22: O ambiente de manipulação de documentos hipermídia

As duas ferramentas são complementares, ou seja, o usuário dispõe da visão estrutural do documento através do browser da estrutura, e das visões temporal e espacial através do browser de sincronismo. O ambiente integrado fornece uma visão completa nos planos de visão estrutural, temporal e espacial de um hiperdocumento.

As três visões são interrelacionadas. Quando um usuário muda o nó em foco no browser da estrutura, a visão no browser de sincronismo é alterada automaticamente, com o posicionamento do nó em foco na origem do eixo do tempo na Time View, passando a ser o objeto base do sincronismo. A mudança do nó em foco, num dado instante, é definida pela seleção explícita no browser da estrutura.

A comunicação entre o browser da estrutura e o browser de sincronismo é feita através de troca de mensagens, indicando, por exemplo, que o nó em foco foi modificado pelo usuário, conforme é detalhado no Capítulo 5. O controle de toda a estrutura dos hiperdocumentos é feita em um módulo independente dos browsers, que através de um protocolo de comunicação permite o compartilhamento de informações entre as duas ferramentas.

Após selecionado o objeto base no browser de sincronismo, o usuário pode transportar outros objetos para definir outras relações síncronas, que também atualizarão o browser da estrutura. Essa operação é realizada selecionando-se um nó no browser da estrutura e informando que este nó deve ser transportado para a visão corrente do browser de sincronismo, seguindo as regras de inclusão citadas na SubSeção 4.2.1.1.

A visão do usuário do ambiente integrado de edição é apresentada na Figura 4.23. Nesse exemplo, o nó em foco pelo usuário é o nó “a1”, que é o objeto base para a definição do sincronismo.

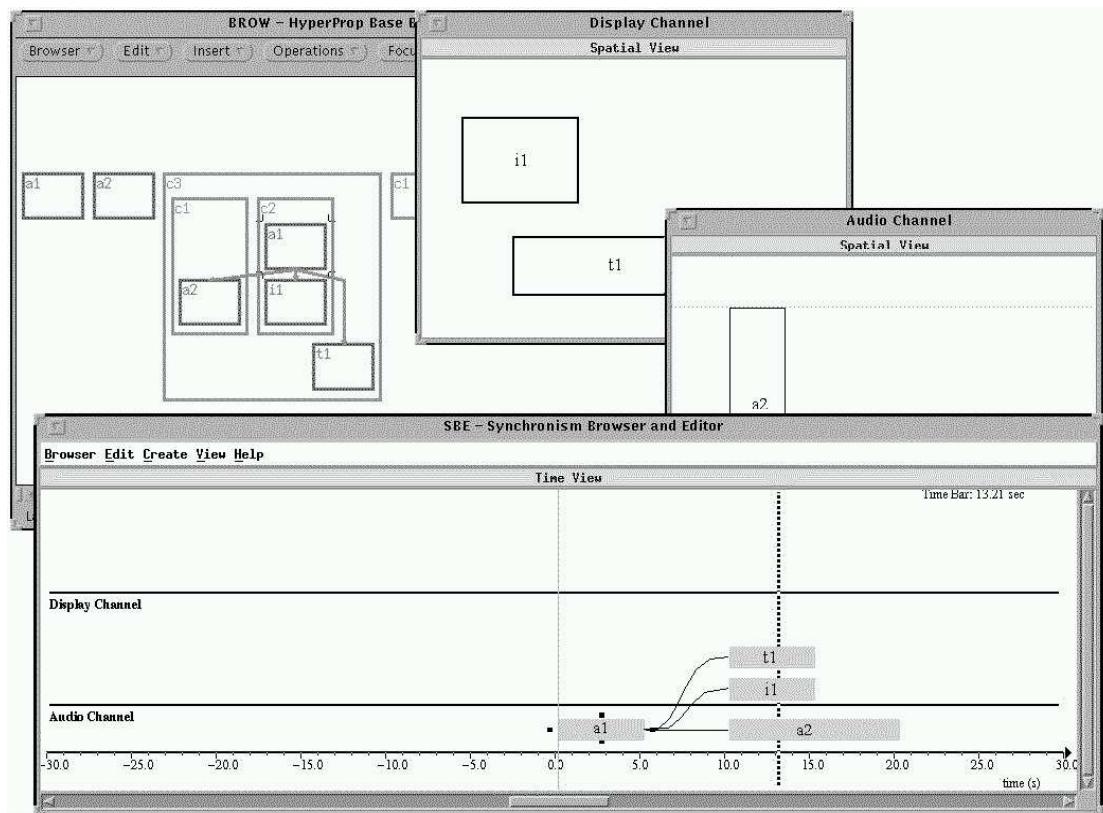


Figura 4.23: Visão do usuário no ambiente integrado de edição e browsing

# Capítulo 5

## Implementação

O Sistema EBS é dividido em cinco módulos: o editor e browser da sincronização temporal, o editor e browser da sincronização espacial, o módulo verificador de consistência, o módulo instanciador de objetos e o módulo de comunicação, conforme pode ser visualizado na Figura 5.1

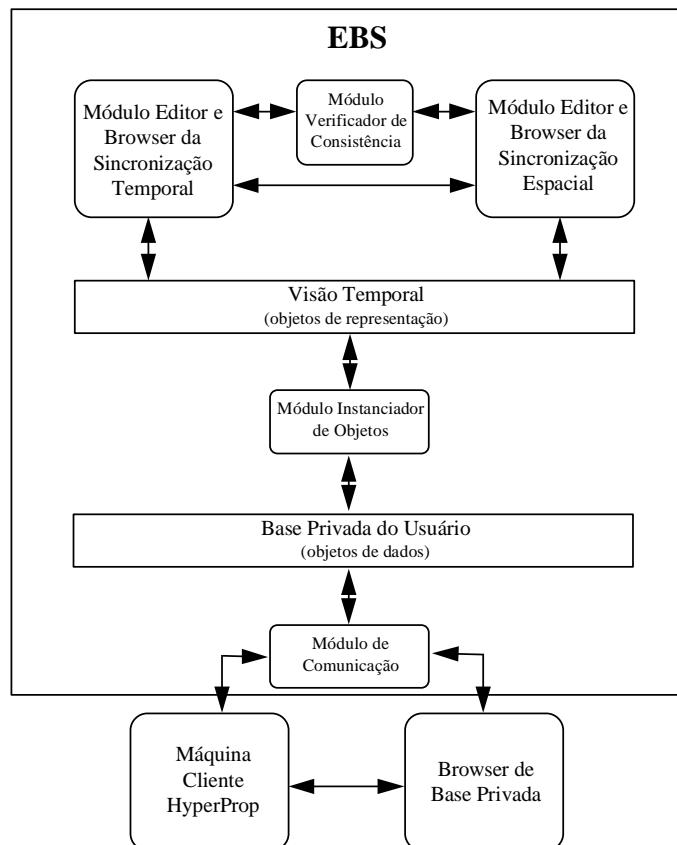


Figura 5.1: Estrutura modular do EBS

Os módulos editores das sincronizações temporal e espacial atuam sobre o plano dos objetos de representação, realizando as funções descritas no Capítulo 4. Todas as interações realizadas pelos usuários nesses módulos passam pela validação do módulo verificador de consistência. O processo de navegação dentro do EBS é representado pela Figura 5.1 pela comunicação direta entre os módulos editores das sincronizações.

O EBS possui um repositório de objetos (base privada do usuário) que corresponde às entidades de dados presentes na base privada do usuário. O módulo instanciador de objetos transforma os objetos de dados da base privada em objetos de representação, que serão utilizados para construir a visão temporal do EBS.

O módulo de comunicação serve para integrar o EBS com o browser de base privada e para realizar as comunicações com a máquina cliente do sistema HyperProp, que coordena a seção de trabalho do usuário.

O EBS funciona de forma cooperativa com o browser de base privada. A interface entre os dois sistemas está compreendida no acesso do EBS a objetos previamente instanciados no browser de base privada, que oferece uma estrutura de dados de acordo com o modelo MCA. O esquema da implementação do ambiente integrado para manipulação de hiperdocumentos é apresentado na Figura 5.2.

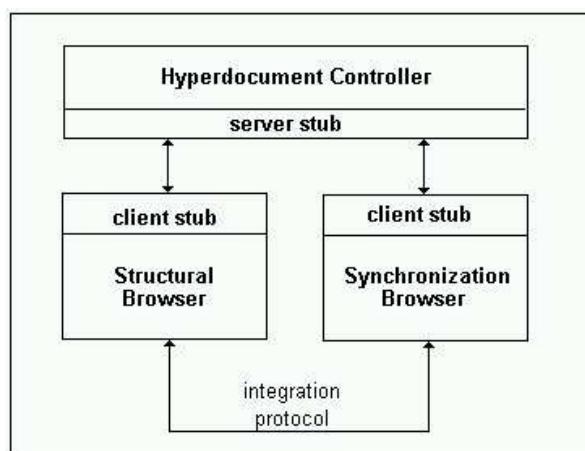


Figura 5.2: Esquema do ambiente integrado de edição e browsing do HyperProp

## 5.1 Requisitos do Sistema

A implementação de um protótipo do EBS é parte integrante da implementação do Modelo de Apresentação do sistema hipermídia HyperProp [Soa95]. Portanto, os requisitos referentes ao Sistema HyperProp são, como consequência, requisitos da implementação da ferramenta.

O sistema deve funcionar independente de plataforma: tipo de máquina, sistema operacional, dispositivos de exibição, etc. A técnica utilizada para o cumprimento deste requisito foi o uso de ferramentas auxiliares que possibilitessem a migração para outras plataformas. Utilizou-se uma biblioteca de rotinas gráficas e uma ferramenta portátil de auxílio à construção de interfaces, denominada IUP/LED [Levy93].

Um dos fatores decisivos para a escolha de uma biblioteca de rotinas gráficas, foi o de se desejar uma implementação o mais portátil possível, de forma independente da plataforma, de maneira a tornar o sistema aberto. O sistema está disponível inicialmente na plataforma UNIX/Motif.

No EBS, deve ser informado inicialmente a plataforma utilizada para exibição, ou seja, qual a configuração da máquina, quais e quantos são os dispositivos para exibição estão disponíveis, etc.

A interface oferecida para o usuário deve ser o mais amigável possível, de maneira a evitar que o autor necessite conhecer detalhes específicos de modelagem de dados e de implementação.

## 5.2 Organização do Programa

Com o objetivo de formalizar a notação, de acordo com o paradigma de orientação a objetos, esta seção utiliza a metodologia de Grady Booch [Booc94],

Na Figura 5.3 é apresentado o diagrama de classes do EBS, que contém uma concepção lógica do sistema. Seu objetivo é mostrar a existência das classes de objetos e suas relações. O diagrama de classes representa a estrutura completa de classes do sistema. A interface com os objetos de dados MCA também é mostrada na figura.

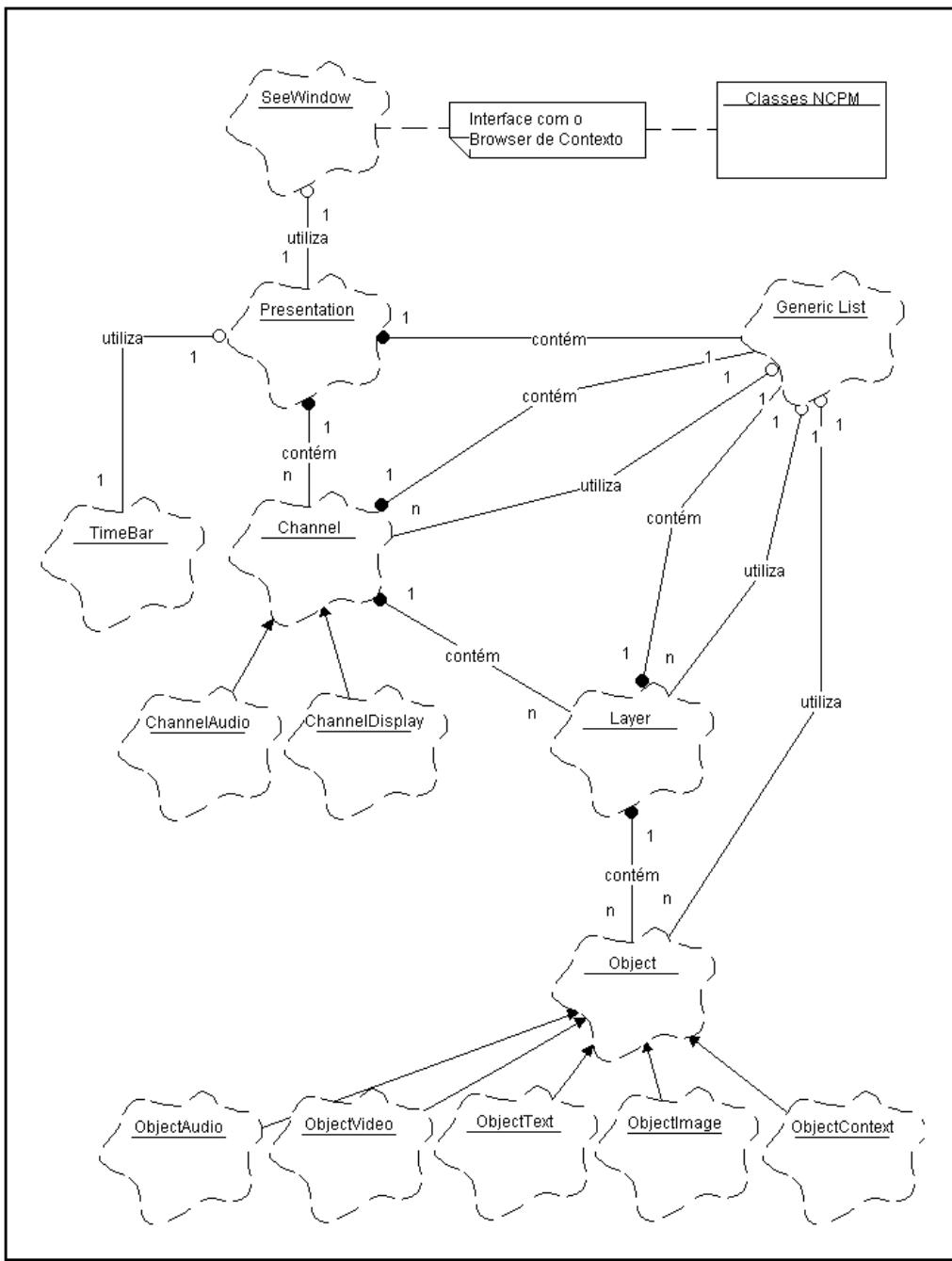


Figura 5.3: A hierarquia de classes do EBS

A seguir são mostrados todos os atributos e métodos das principais classes do EBS. A Figura 5.4 ilustra a notação utilizada na descrição das classes.

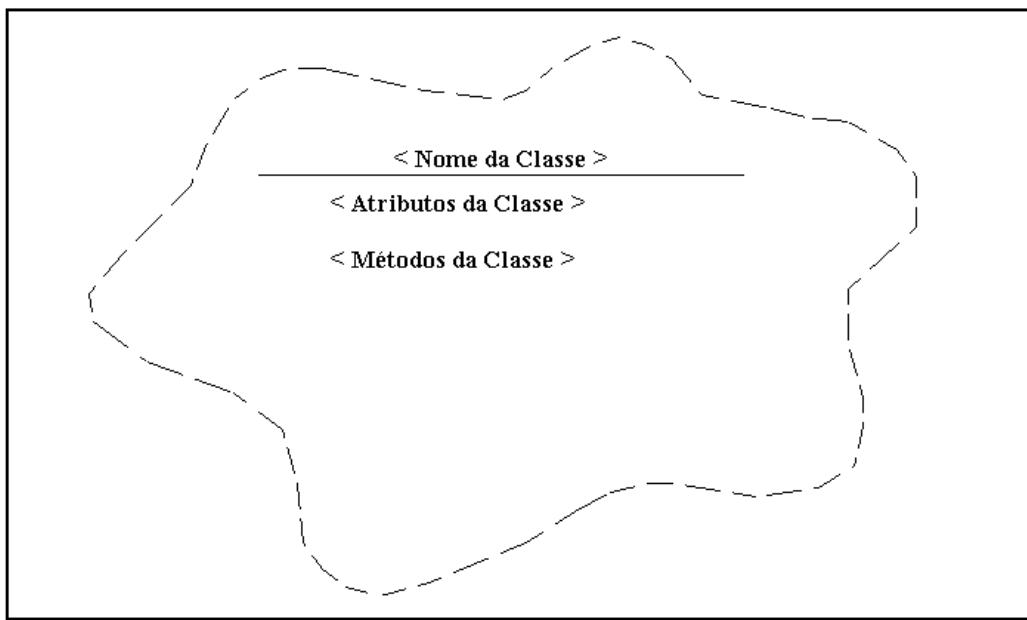


Figura 5.4: Notação para descrição das classes

A Figura 5.5 mostra os atributos e os métodos da classe SeeWindow, que tem a finalidade de gerenciamento das janelas do sistema.

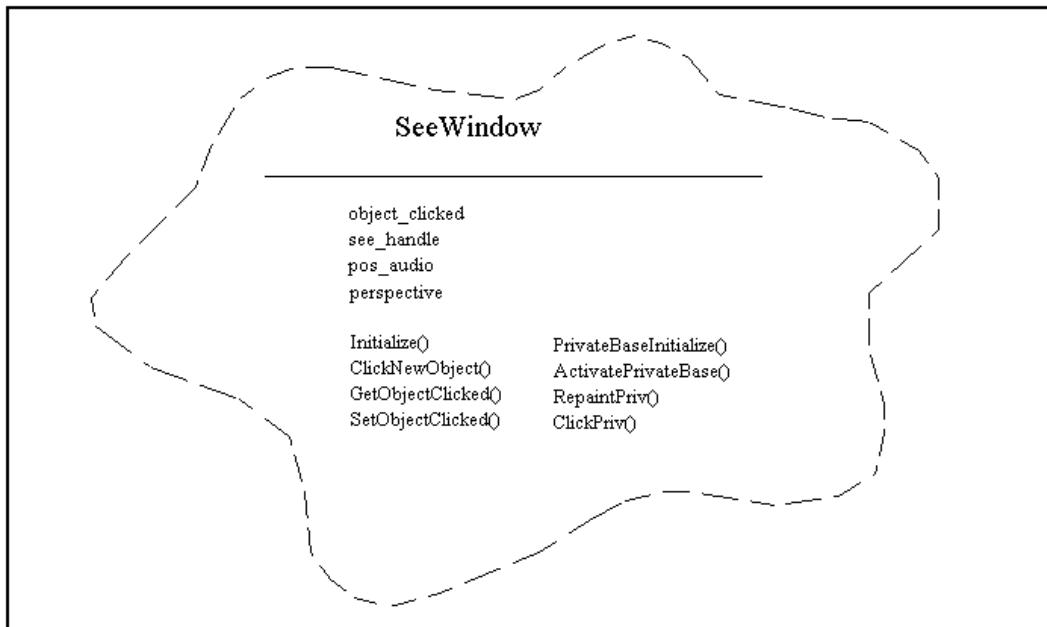


Figura 5.5: A classe SeeWindow

A Figura 5.6 mostra os atributos e os métodos da classe Presentation. A classe Presentation é a principal entidade do sistema. A barra de tempo da Time View também é modelada como uma entidade pertencente a classe Presentation.

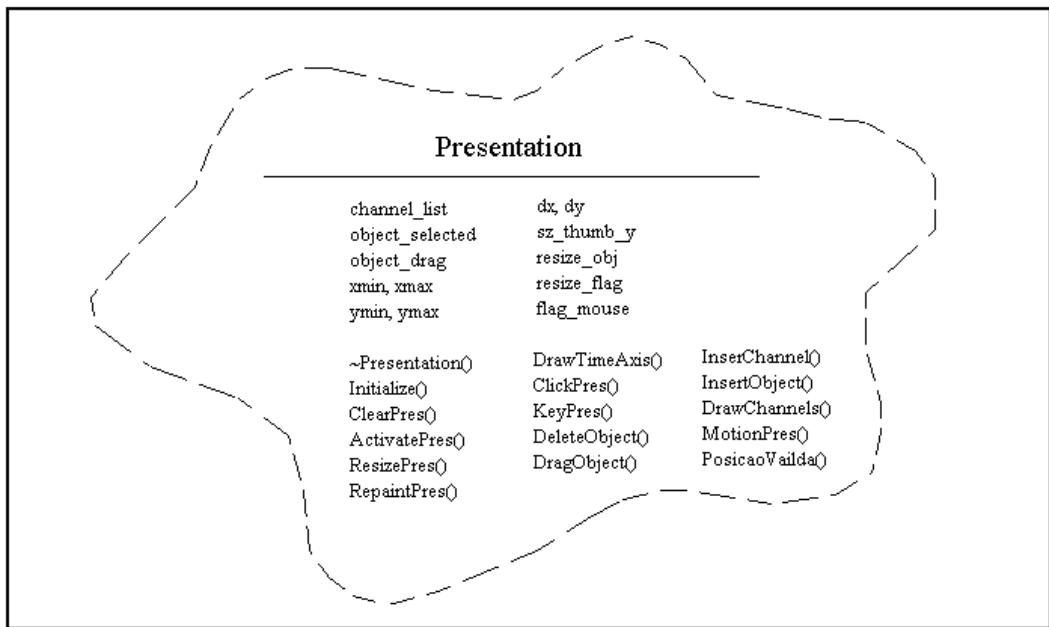


Figura 5.6: A classe Presentation

A Figura 5.7 mostra os atributos e os métodos da classe Channel. A classe Presentation possui uma lista de objetos da classe Channel, que correspondem aos canais de dispositivos da plataforma de exibição.

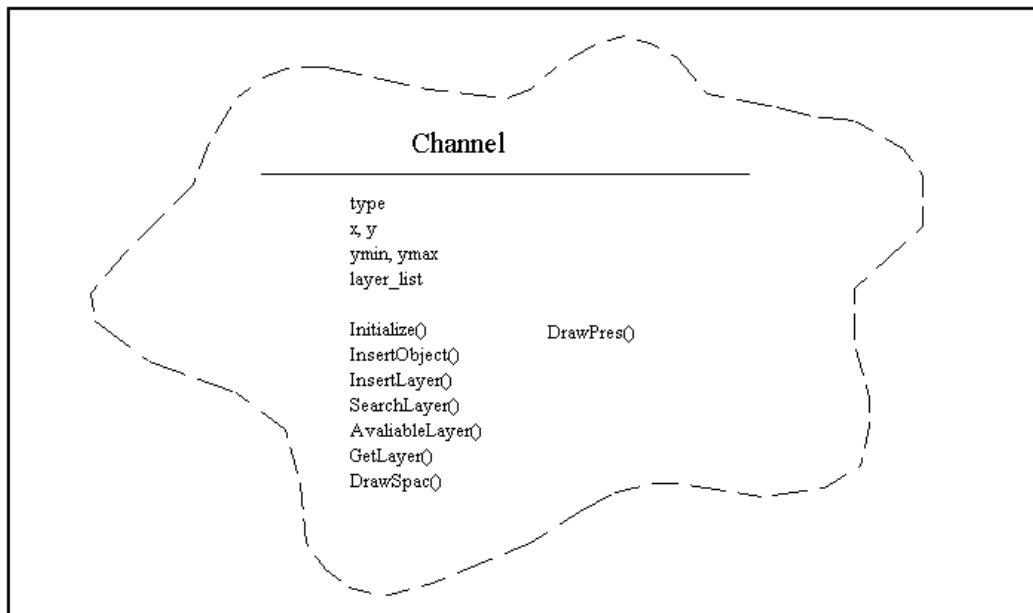


Figura 5.7: A classe Channel

A Figura 5.8 mostra os atributos e os métodos da classe Layer. A classe Channel possui uma lista de objetos da classe Layer, que indicam a ordem da sobreposição dos objetos na Spatial View.

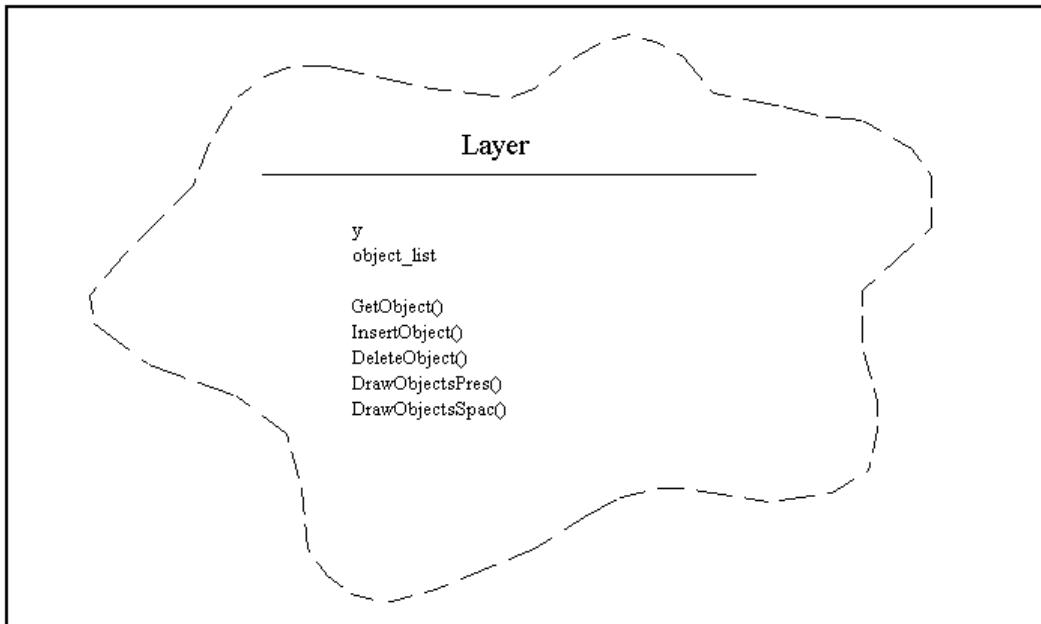


Figura 5.8: A classe Layer

A Figura 5.9 mostra os atributos e os métodos da classe Object. A classe Layer possui uma lista de objetos da classe Object, que contém a informação dos objetos de dados. A classe Object é subdividida nas subclasses correspondentes as diversas mídias.

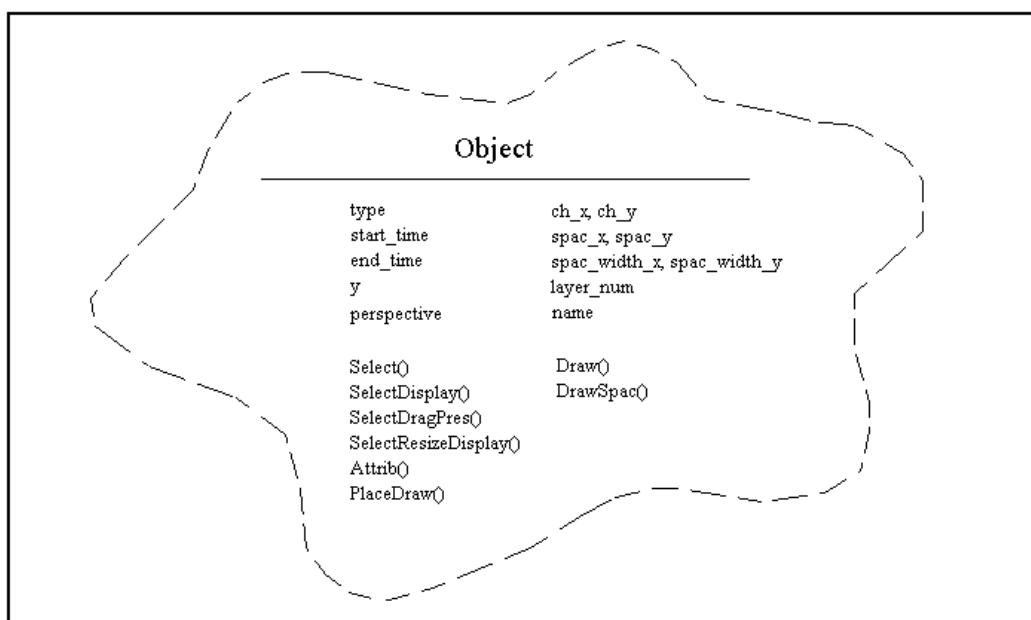


Figura 5.9: A classe Object

## 5.3 Testes

Foram adotados alguns critérios para a realização de testes sistemáticos no programa. Devido ao grande número de classes e métodos utilizados, optou-se por realizar a metodologia bottom-up para realizar testes em funções específicas na medida em que elas foram construídas.

Foi utilizado o *teste da caixa preta*, [Rumb91], que enfoca os requisitos funcionais de um método. Através desse teste foi possível detectar o cumprimento dos requisitos apresentados na Seção 5.1. O teste da caixa preta visa detectar:

- a especificação incorreta de funções;
- erros de interface;
- erros na estrutura de dados utilizada;
- erros de inicialização e terminação;

Foram selecionadas entradas válidas e inválidas baseadas nos requisitos especificados, com o propósito de observar os efeitos produzidos.

O programa possui a característica de verificar continuamente a consistência dos dados fornecidos pelo usuário, facilitando a análise dos resultados dos testes realizados. A utilização de depuradores de código também foi uma técnica válida para testar alguns módulos do sistema.

## 5.4 A Interface do EBS no Ambiente UNIX/Motif

A seguir é mostrada a interface do sistema EBS implementado sobre a plataforma UNIX/Motif. Na Figura 5.10 é mostrada a janela Time View. As Figuras 5.11 e 5.12 mostram as janelas que compõem a Spatial View, a janela de Canal de Display (*display channel*) e Canal de Áudio (*audio channel*), respectivamente.

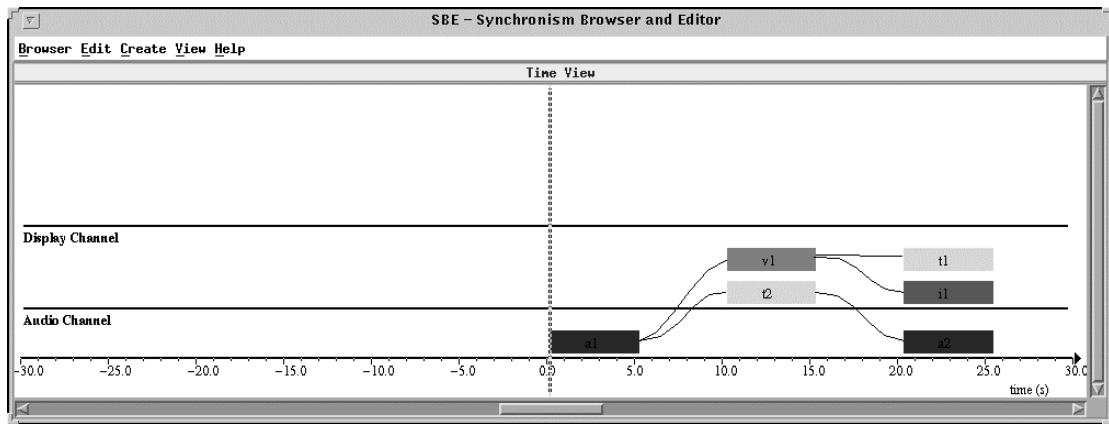


Figura 5.10: A janela Time View no ambiente UNIX/Motif

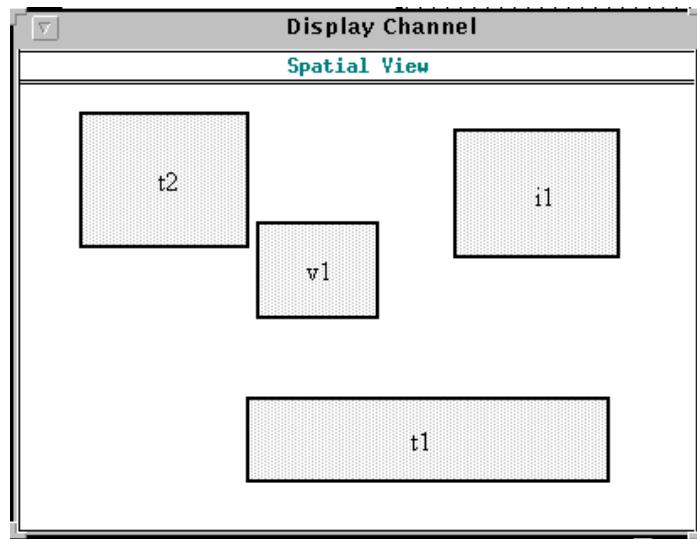


Figura 5.11: A janela canal de display na Spatial View no ambiente UNIX/Motif

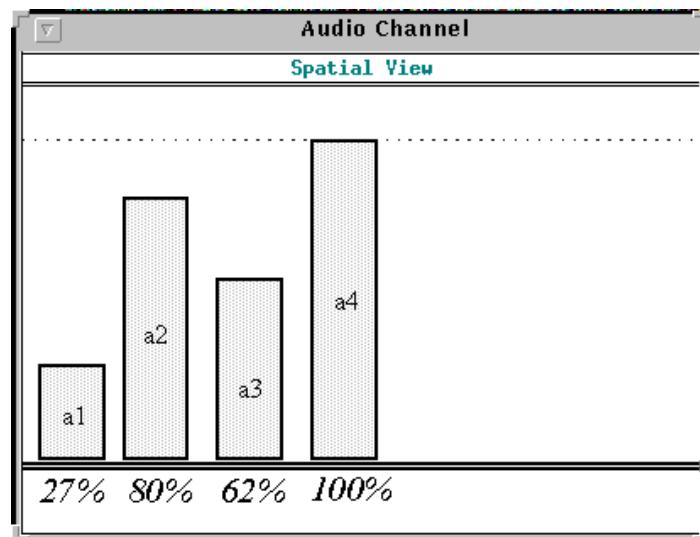


Figura 5.12: A janela canal de áudio na Spatial View no ambiente UNIX/Motif

# Capítulo 6

## Conclusões

Um protótipo do EBS se encontra hoje disponível e em fase de teste e uso. Esse protótipo, desenvolvido no âmbito do Projeto HyperProp no Laboratório TeleMídia da PUC-Rio, foi implementado como mencionado no capítulo anterior em ambiente UNIX/Motif, usando a linguagem de programação C++ e uma ferramenta portátil de auxílio à construção de interfaces. A utilização dessa ferramenta visou tornar o EBS um sistema multi-plataforma.

Dos testes em realização, espera-se ganhar experiência sobre o comportamento da interface na prática e as necessidades que possam surgir durante a utilização do sistema.

O objetivo da implementação da ferramenta foi validar os aspectos de sincronização do modelo conceitual e realizar uma interação entre o EBS e o browser de base privada, incluindo os mecanismos de filtragem, conforme descrito em [Much96]. O resultado obtido foi uma integração total do editor e exibidor das estruturas, descrito naquela referência, com o EBS.

Na Seção 6.1 são realizadas comparações entre o EBS e os demais sistemas de edição de sincronização apresentados no Capítulo 2. A Seção 6.2 apresenta as principais contribuições desta dissertação. Na Seção 6.3 são propostos trabalhos futuros que

podem ser realizados como complemento à vários aspectos abordados nesta dissertação.

## 6.1 Comparação com Trabalhos Relacionados

Nesta seção é realizada uma comparação entre o EBS e os trabalhos apresentados no Capítulo 2. Os critérios que serão utilizados para analisar os sistemas são:

- paradigma utilizado - indica qual o paradigma utilizado na implementação do sistema (ver Seção 2.2).
- tipos de sincronização - indica se o sistema possui mecanismos para realizar as sincronizações temporal e/ou espacial.
- tipo de sincronização temporal - indica se os componentes de um documento encontram-se inter-relacionados temporalmente (sincronização relativa), ou se não há interrelacionamento temporal entre os componentes (sincronização absoluta).
- tipo de sincronização espacial - indica como é realizada a sincronização espacial.
- relacionamentos temporais - define a granularidade das relações temporais.
- alteração de comportamento - indica se, no sistema, o autor é capaz de definir alteração de comportamento espacial nos componentes de um documento.
- manutenção da consistência temporal - indica se o sistema é capaz de manter a consistência temporal especificada pelo autor durante o processo de edição da apresentação de um documento.
- interface gráfica - indica se o sistema possui interface gráfica para a definição da sincronização temporal/espacial.
- simulação da apresentação - indica se o sistema é capaz de realizar simulações da apresentação durante o processo de edição da apresentação de um documento

A Tabela 6.1 mostra as características dos sistemas segundo a análise dos critérios acima apresentados.

	<i>CMIFed</i>	<i>Firefly</i>	<i>Videobook</i>	<i>Toolbook</i>	<i>Director</i>	<i>MAEstro</i>	<i>EBS</i>
<b>paradigma utilizado</b>	baseado em eventos	baseado em eventos	scripting	scripting	scripting e timeline	timeline	baseado em eventos
<b>tipos de sincronização</b>	temporal e espacial	temporal e espacial	temporal e espacial	temporal e espacial	temporal e espacial	temporal	temporal e espacial
<b>sincronização temporal</b>	relativa	relativa	relativa	relativa	relativa e absoluta	absoluta	relativa
<b>sincronização espacial</b>	relativa	relativa	relativa	via scriptware	via scriptware	—	relativa
<b>relacionamentos temporais</b>	relação 1:1	relação 1:1	relação 1:1	relação m:n ( script )	relação m:n ( script )	relação 1:1	relação 1:1
<b>alteração de comportamento</b>	sim	sim	não	não	não	não	sim
<b>mantém consistência temporal ?</b>	sim	sim	sim	não	não	não	sim
<b>interface gráfica</b>	temporal e espacial	temporal e espacial	temporal e espacial	espacial	temporal e espacial	temporal	temporal e espacial
<b>realiza simulação da apresentação ?</b>	sim	sim	não	sim	sim	sim	não

Tabela 6.1: Comparação entre alguns sistemas de edição de sincronização

O EBS utiliza o paradigma baseado em eventos para relacionar temporalmente os componentes de um hiperdocumento, conforme discutido no Capítulo 3. O sincronismo da apresentação é realizado através do relacionamento dos eventos. O benefício dos sistemas que utilizam esse paradigma (EBS, CMIFed e Firefly), é que todos os componentes de um documento que possuem algum tipo de relacionamento síncrono são dependentes uns dos outros, diferentemente da técnica de timeline usada nos sistemas MAEstro e Director, onde uma simples alteração num segmento da

apresentação pode requerer que todos os tempos definidos para eventos posteriores tenham que ser atualizados manualmente, pois não há interrelacionamento entre os componentes do documento.

Uma desvantagem dos sistemas que utilizam a técnica de scripting (Toolbook, Director), é que a programação e manutenção de apresentações de grande porte são tarefas complexas. Torna-se difícil visualizar a estrutura temporal de documentos que utilizam linguagens scriptware. Uma exceção é o sistema Videobook que utiliza scripts visuais, conforme apresentado na Seção 2.5.

## 6.2 Principais Contribuições

A grande motivação desta dissertação foi estudar mecanismos de auxílio à definição da sincronização temporal e espacial em sistemas hipermídia, concentrando em uma ferramenta gráfica que visasse facilitar essa tarefa. Nesta dissertação foi implementado um sistema que auxilia o usuário no processo de edição e exibição do sincronismo de hiperdocumentos, o EBS.

Muitas técnicas de sincronização foram desenvolvidas assumindo que a maioria dos sistemas existentes seguiam paradigmas que dificultavam a interação do usuário e, principalmente, sistemas difíceis de se utilizar. O EBS é um sistema que utiliza o paradigma baseado em eventos para relacionar temporalmente os componentes de um documento, onde os objetos se relacionam entre si, tornando o sincronismo temporal relativo e, consequentemente, mais fácil de ser utilizado.

A ferramenta possui um mecanismo de verificação da consistência da definição da sincronização temporal de objetos, realizando um processo de compilação incremental que avalia a especificação construída pelo usuário a cada interação.

No EBS foi implementado um mecanismo onde o autor pode criar relações temporais de alto nível entre os componentes de um documento, facilitando o processo de edição da sincronização temporal.

A implementação do EBS possibilitou a construção de uma ferramenta única de edição e *browsing* para o Sistema HyperProp totalmente integrada, formada pelo browser de base privada, que fornece a visão estrutural de um documento, e pelo EBS, que fornece as visões temporal e espacial.

A implementação do EBS foi realizada no plano dos objetos de representação, permitindo que a apresentação seja composta por instâncias dos objetos de dados inter-relacionadas temporal e espacialmente.

### **6.3 Trabalhos Futuros**

A implementação atual do EBS trata apenas os elos de sincronização que possuem relação *um para um* (1:1). Um trabalho que deve ser realizado é a implementação de relacionamentos *muito-para-muitos* (m:n) nos elos de sincronização.

Outro trabalho que deve ser realizado é a implementação de um mecanismo que permita realizar simulações na exibição dos hiperdocumentos durante o processo de edição. A possibilidade de se fazer simulações da apresentação certamente daria ao autor um retorno significativo da sincronização do documento que está sendo editado.

A integração do EBS com o módulo de controle de execução do sistema HyperProp também se constitui em um trabalho necessário. Essa integração garantirá que a sincronização temporal e espacial dos hiperdocumentos durante suas exibições siga de acordo com a especificação realizada pelo autor no EBS.

Torna-se necessário prover o EBS de técnicas que permitam realizar o processo de compilação incremental para verificação da consistência temporal dos objetos de maneira mais rápida, com o objetivo de aumentar a interatividade do autor com a ferramenta.

O EBS foi inicialmente projetado para validar as especificações do subsistema de apresentação do HyperProp, sendo utilizado na prática por usuários que conhecem a modelagem dos dados manipulados pela ferramenta. Entretanto, torna-se necessário avaliar aspectos de usabilidade e adequação da interface gráfica levando-se em

consideração sua adequação ao usuário final. Técnicas específicas de avaliação de interfaces precisam ser utilizadas com tal finalidade.

O EBS e as outras ferramentas do sistema HyperProp necessitam de um mecanismo eficiente para se atualizar dinamicamente, caso ocorram alterações nas entidades de dados instanciadas nas bases privadas dos usuários.

# Referências Bibliográficas

- [Alle83] Allen, J. F., *Maintaining Knowledge About Temporal Intervals*, Communications of the ACM, 26(11):832-843, Novembro de 1983.
- [Alle84] Allen, J. F., *Towards a General Theory of Action and Time*, Artificial Intelligence, 23:123-154, 1984.
- [AlFe94] Allen, J. F.; Ferguson, G., *Actions and Events in Interval Temporal Logic*, Technical Report, University of Rochester, New York, Julho de 1994.
- [Bati94] Batista, T., *Um Sistema de Autoria para Hiperdocumentos*, Dissertação e Mestrado, Departamento de Informática, PUC-Rio, March, 1994
- [Booc94] Booch, G., *Object Oriented Analysis and Design with Applications*; second edition; Addison Wesley, 1994.
- [BuZe92] Buchanan, M.C.; Zellweger, P.T., *Specifying Temporal Behavior in Hypermedia Documents*, Proceedings of European Conference on Hypertext, ECHT'92, Milano, December 1992.
- [BuZ93a] Buchanan, M.C.; Zellweger, P.T., *Automatically Generating Consistent Schedules for Multimedia Documents*, Multimedia Systems J., Springer-Verlag, April 1993.
- [BuZ93b] Buchanan, M.C.; Zellweger, P.T., *Automatic Temporal Layout Mechanisms*, Proceedings of ACM Multimedia'93, Anaheim, CA. August 1993.
- [Cost95] Costa, F.R., *Um Ambiente para Edição de Sincronismo em Hiperdocumentos*, Projeto Final de Programação, Departamento de Informática, PUC-Rio, Dezembro 1995.
- [Cost96] Costa, F.R., *Editor e Browser Gráfico para Sincronização Temporal e Espacial de Objetos Multimídia/Hipermídia*, Anais do II Workshop em Sistema Hipermídia Distribuídos, Fortaleza, Ceará, Maio 1996.
- [DrGr91] Drapeau, G. D.; Greenfield, H., *MAEstro - A Distributed Multimedia Authoring Environment*, Proceedings of USENIX Summer 1991, June 1991.

- [DrGr93] Drapeau, G. D.; Greenfield, H., *Synchronization in the MAEstro Multimedia Authoring Environment*, Parallax Graphics, Inc., Santa Clara, CA., 1993.
- [Furn86] Furnas, G., *Generalized Fisheye Views*, Proceedings of CHI'86 Human Factors in Computing Systems, Boston, April, pp. 16-23, 1986.
- [HaSc90] Halasz, F.; Schwartz, M., *The Dexter Hypermedia Reference Model*, NIST, Hypertext Standardization Workshop, Gaithersburg, MD, January 1990, 16-18.
- [HRB93a] Hardman, L.; van Rossum, G.; Bulterman, D. C. A., *The Amsterdam Hypermedia Model: extending hypertext to support real multimedia*, Hypermedia, May 1993, 5(1).
- [HRB93b] Hardman, L.; van Rossum, G.; Bulterman, D. C. A., *Structured Multimedia Authoring*, Proceedings of ACM Multimedia'93, Anaheim, CA. August 1993.
- [Levy93] Levy, C. H., *IUP/LED: Uma Ferramenta Portátil de Interface com o Usuário*, Dissertação de Mestrado, Departamento de Informática, PUC-Rio, 1993.
- [MacD89] *MacroMind Director: Overview Manual*, MacroMind Inc., March 1989.
- [MuSo95] Muchaluat, D.C.; Soares, L.F.G., *Browsers e Trilhas no Sistema HyperProp*, Anais do I Workshop em Sistemas Hipermídia Distribuídos, USP, São Carlos, Julho, 1995
- [MuSC95] Muchaluat, D.C.; Soares, L.F.G.; Casanova, M.A., *Browsing in a Hypermedia System with Nested Composite Nodes*, Relatório Técnico, Departamento de Informática, PUC-Rio, October, 1995
- [Much96] Muchaluat, D.C., *Browsers e Trilhas para Documentos Hipermídia Baseados em Modelos com Composições Aninhadas*, Dissertação de Mestrado, Departamento de Informática, PUC-Rio, 1996.
- [NetProg] *Networking Programming Guide*, Sun Microsystems, Inc., Mountain View, CA.
- [OgHK90] Ogawa, R.; Harada, H.; Kaneko, A., *Scenario-based Hypermedia: A Model and a System*, Proceedings of European Conference on Hypertext, ECHT'90, November 1990, INRIA France, 38-51.
- [RJMB93] van Rossum, G.; Jansen, J.; Mullender, K.S. and Bulterman D., *CMIFed: A Presentation Environment for Portable Hypermedia Documents*, Proceedings ACM Multimedia'93. Anaheim, CA. August 1993.

- [Rumb91] Rumbaugh, J. & alii, *Object-Oriented Modeling and Design*. Ed. Prentice Hall, New York, 1991.
- [SaBr92] Sarkar, M.; Brown, M., *Graphical Fisheye View of Graphs*, Proceedings of CHI'92 Human Factors in Computing Systems, pp. 83-91, May, 1992
- [Soar95] Soares, L.F.G. et alii, *HyperProp: Uma Visão Geral*, Anais do I Workshop em Sistema Hipermídia Distribuídos, São Carlos, São Paulo, Julho 1995.
- [SoCa93] Soares, L.F.G.; Casanova, M.A., *Modelo de Contextos Aninhados com Intercâmbio de Objetos MHEG em Arquiteturas Distribuídas*, Anais do XI Simpósio Brasileiro de Redes de Computadores, Campinas, São Paulo, Maio, 1993
- [SoCC93] Soares, L.F.G.; Casanova, M.A.; Colcher, S., *An Architecture for Hypermedia Systems Using MHEG Standards Object Interchange*, Proceedings of the Workshop on Hypermedia and Hypertext Standards, Amsterdam, The Netherlands, April, 1993
- [SoRC93] Soares, L.F.G.; Rodriguez, N.L.R.; Casanova, M.A., *An Open Hypermidia System with Nested Composite Nodes and Version Control*, Departamento de Informática PUC-Rio, December, 1993
- [SoCR94] Soares, L.F.G.; Casanova, M.A.; Rodriguez, N.L.R., *Modelo de Contextos Aninhados: um Modelo Conceitual Hipermídia*. Revista Brasileira de Computação, 7(2), Janeiro 1994.
- [SoSC95] Souza, G.L.; Soares, L.F.G.; Casanova M.A., *Synchronization Aspects of an Hypermedia Presentation Model with Composite Nodes*, Proceedings of the ACM Worshop on Effective Abstractions in Multimedia, in connection with ACM Multimedia 95, São Francisco, EUA, Novembro 1995.
- [SoSo96] Souza, G.L.; Soares, L.F.G., *Edição e Execução de Aplicações de Documentos Hipermídia no HyperProp*, Anais do II Workshop em Sistema Hipermídia Distribuídos, Fortaleza, Ceará, Maio 1996.