Editor Gráfico para Estrutura e Sincronismo de Documentos Multimídia

Fábio Rodrigues Costa[1] fcosta@inf.puc-rio.br
Débora C. Muchaluat[1] debora@inf.puc-rio.br
Luiz Fernando G. Soares[1] lfgs@inf.puc-rio.br
Guido L. de Souza Filho[1,2] guido@inf.puc-rio.br

[1] Laboratório TeleMídia - Depto. de Informática, PUC-Rio
R. Marquês de São Vicente 225
22453-900 - Rio de Janeiro, Brasil

[2] DIMAp, UFRN
Campus Universitário, Lagoa Nova
59072-970 - Natal - RN, Brasil

Abstract

This paper presents a graphic editor for the definition of multimedia documents' structure and synchronization. The tool offers three different views for documents. The first view manipulates their structure, the second one treats their temporal relationships and the third one specifies their spatial characteristics. We discuss the need for these views, relating them to a conceptual model and to an open architecture for document handling. Finally, we present an implementation of the tool in an editor built over the HyperProp multimedia/hypermedia system.

1 - Introdução

Sistemas multimídia devem facilitar o trabalho de autoria oferecendo um ambiente completo para edição e manipulação de documentos. Editores gráficos para a definição da estrutura de documentos, definição das relações de sincronização entre seus componentes e especificação de mudanças no seu comportamento durante uma apresentação multimídia são cruciais para esses sistemas. Este artigo descreve parte do processo de autoria de documentos multimídia através de três diferentes visões, como apresentado na Figura 1.

A primeira visão considera a edição da estrutura do documento, fornecendo recursos para editar fragmentos de informação (nós), seus relacionamentos, e o agrupamento desses nós em composições. Essa visão é apresentada no plano superior da Figura 1, onde fragmentos de informação são representados por retângulos, e o relacionamento de inclusão em composições é representado pela inclusão de um retângulo (nó) em outro retângulo (nó de composição). Composições aninhadas são facilidades obrigatórias em modelos conceituais multimídia/hipermídia mais recentes, exemplificadas no próprio padrão MHEG [MHEG95].

A segunda visão é responsável pela especificação dos relacionamentos temporais entre os componentes de um documento multimídia, definindo suas posições relativas no tempo, como exibido no plano da esquerda da Figura 1. Nesse plano, nós são representados por retângulos, cujos comprimentos indicam suas durações de exibição e cujas posições relativas definem suas sincronizações no tempo.

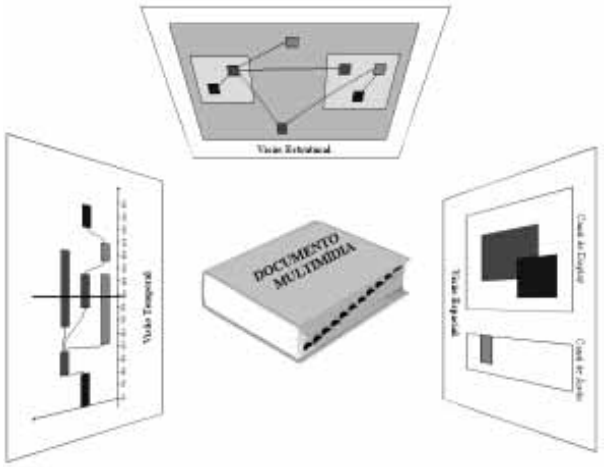


Figura 1 - Diferentes visões de um documento multimídia.

Finalmente, a terceira visão permite a definição de relacionamentos espaciais entre componentes de um documento multimídia, estabelecendo suas características de apresentação em um determinado dispositivo, em um dado instante do tempo, como apresentado no plano da direita da Figura 1. Nesse plano, os retângulos representam as características espaciais dos objetos em um dispositivo de exibição, especificando sua posição em um monitor de vídeo, seu volume em um dispositivo de áudio, etc.

No processo apresentado, as três visões estão relacionadas, de modo que um objeto em foco na visão estrutural é a base para a seqüência de sincronismo mostrada na visão temporal. Nessa última visão, para cada ponto selecionado no tempo, a visão espacial mostra

como os componentes serão apresentados no espaço definido pelos dispositivos de saída.

O principal propósito deste artigo é apresentar um ambiente integrado para navegação e edição oferecendo as visões anteriormente apresentadas. Na Seção 2, são discutidos alguns conceitos relevantes para o tratamento de documentos em sistemas que oferecem uma arquitetura aberta em camadas, posicionando os editores gráficos de acordo com objetos do modelo conceitual considerado. Na Seção 3, a ferramenta gráfica integrada para a edição da estrutura e sincronismo de documentos multimídia é apresentada. Finalmente, a Seção 4 é reservada às conclusões. Trabalhos relacionados são mencionados no decorrer do texto.

2 - Documentos Multimídia com Composições Aninhadas

Documentos multimídia são constituídos por fragmentos de informação representados em várias mídias, denominados *nós*. A estrutura de um documento pode ser definida através de uma hierarquia de *composições*, onde os nós são agrupados em nós de composição, que podem conter outras composições, recursivamente.

Em algumas aplicações, as informações multimídia (nós) estão relacionadas através de ligações denominadas *elos*. Os elos, nos modelos conceituais mais recentes, incluindo o adotado neste trabalho [SoCR95], são definidos nos nós de composição e permitem a definição de relações de sincronismo, entre outras. Elos são caracterizados como uma relação m:n entre dois conjuntos de extremidades (origem e destino) associados a um ou mais nós.

A Figura 2 apresenta uma arquitetura aberta para o tratamento de documentos multimídia/hipermídia em conformidade como Modelo Dexter [HaSc90] e a proposta MHEG [MHEG95].

A *camada de armazenamento* implementa o armazenamento persistente de objetos multimídia/hipermídia e oferece uma interface para o intercâmbio de dados, chamada *multimedia hypermedia interchangeable objects* (interface MHIO).

A interface MHIO é a chave para fornecer a compatibilidade entre aplicações e equipamentos, pois estabelece dois pontos em que as camadas de armazenamento e apresentação devem concordar: (1) a representação codificada para objetos multimídia a serem intercambiados, correspondente ao padrão ISO MHEG; e (2) as mensagens, pedidos, confirmações etc., usados por essas camadas para solicitar um objeto, conteúdo ou ação requisitados por camadas superiores.

Figura 2 - Arquitetura Multimídia/Hipermídia em Camadas

A *camada de aplicação* introduz os conceitos de *objetos de dados* e de *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, exceto o 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. Objetos de representação oferecem diferentes visões de objetos de dados, como será discutido mais adiante.

A principal função da *camada de apresentação* é converter de/para o formato dos objetos de dados usados por aplicações e plataformas particulares e a representação codificada para objetos multimídia definida pela interface MHIO. Maiores detalhes sobre a arquitetura podem ser encontrados em [SoCC93].

A informação necessária para apresentar os componentes de um documento multimídia é fornecida pelo objeto de representação, formado pela associação de um objeto de dados e um objeto da classe descritor. Um *descritor* especifica como um objeto de dados será exibido para o usuário, detalhando como 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 em listas de operações, como em [BuZe92]. A associação entre objetos de dados e descritores é representada na Figura 3 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 ar-

cos em círculos maiores. Os planos apresentados podem ser modelados por nós de composição, em concordância com o modelo.

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 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, conforme detalhado em [SoSC95].

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 editor de estrutura, apresentado na Seção 3.1, atua no plano de objetos de dados, uma vez que toda a informação necessária encontra-se nesse plano. Por outro lado, os relacionamentos de sincronismo possuem apenas significado no plano de representação. Assim, é nesse plano que o editor de sincronismo temporal e espacial atuará, como descrito na Seção 3.2.

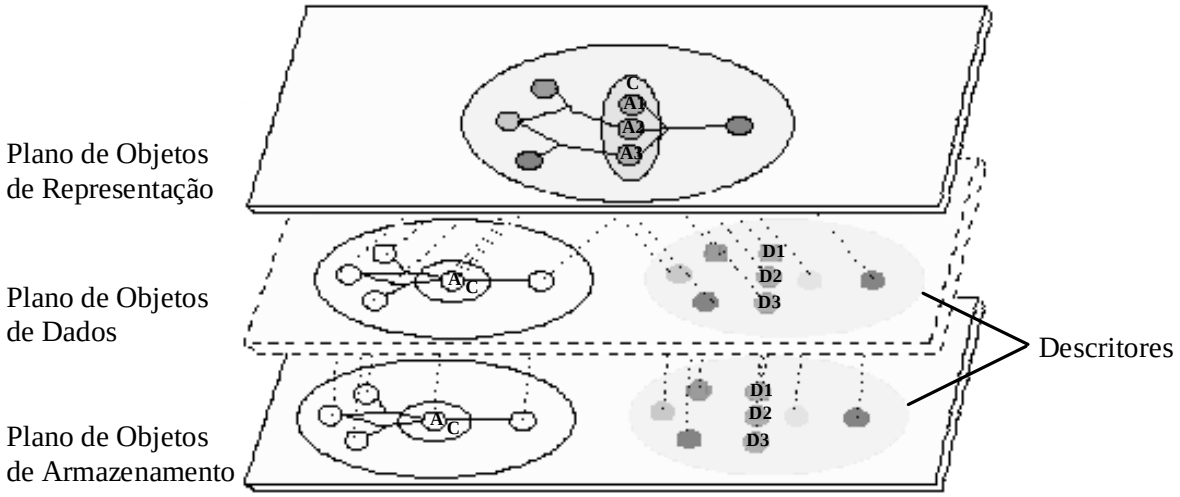


Figura 3 - Planos de armazenamento, de dados e de representação.

3 - Editor Gráfico para a Manipulação de Documentos Multimídia

A interface do editor gráfico é ilustrada na Figura 4. Cada janela do editor corresponde a uma visão para a manipulação de documentos. A janela do canto superior esquerdo fornece a visão estrutural do documento, onde os nós são representados por retângulos incluídos em outros retângulos (nós de composição), como anteriormente mencionado. A janela inferior apresenta a visão temporal do documento e a janela do canto superior direito apresenta a visão espacial do documento em um determinado instante de tempo. As visões trabalham de forma cooperativa, permitindo ao usuário editar a estrutura de um documento multimídia e especificar suas características de apresentação temporais e espaciais de forma simultânea, se assim desejar.

Na figura, parte de um documento sobre os "Beatles" é exibido como exemplo. No documento, o nó em foco selecionado na visão estrutural é um áudio que contém a canção "Imagine". A time view mostra a *time chain* (cadeia de síncrona de nós interconectados por elos) baseada nesse nó. A barra de tempo está posicionada em 5 segundos após o início da *time chain*. Nesse instante relativo do tempo, o autor especificou que devem ser apresentados a letra da música (nó de texto ImagineLyrics), uma foto de John Lenon (nó de imagem Lenon) e um clip sobre os Beatles (nó de vídeo BeatlesClip). A visão espacial mostra como os nós usam os dispositivos de saída nesse instante.

As subseções que se seguem detalham as principais características de cada visão.

3.1 - Visão Estrutural

No caso de um editor de estrutura para documentos com composições que se deseja visualizar com detalhes em todos os níveis de aninhamento, pode-se obter um diagrama bastante complexo, requerendo métodos sofisticados para a construção da ferramenta. Filtros para esconder informações, e possivelmente landmarks (nós especiais que sempre aparecem) serão necessários.

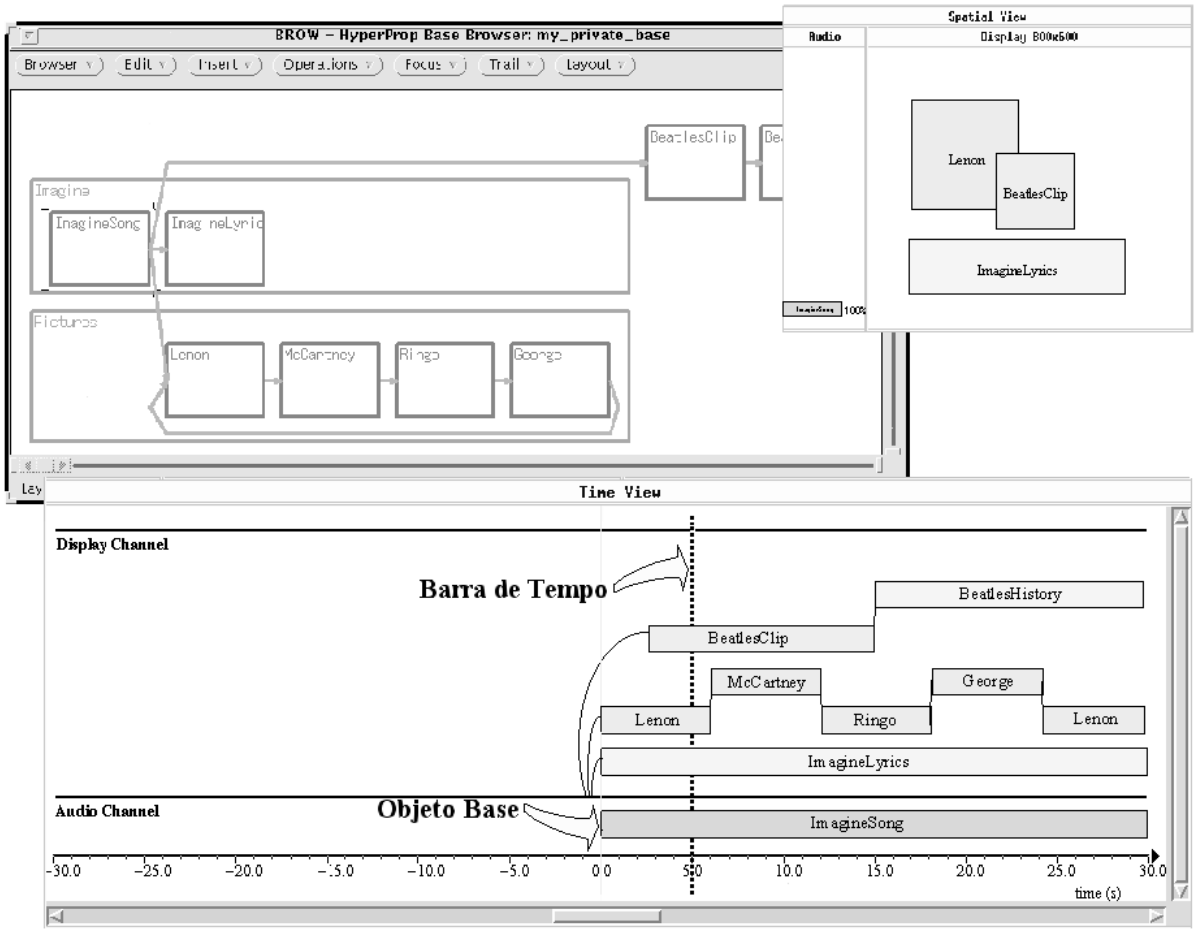


Figura 4 - interface do editor gráfico para a manipulação de documentos multimídia

Manter a legibilidade dos mapas é o principal objetivo ao se usar filtros para esconder informações. Para a construção desses filtros, foi utilizada uma extensão da estratégia de visões olho-de-peixe para modelos conceituais que oferecem nós de composição [Much96].

A motivação fundamental da estratégia de olho-de-peixe para documentos [Furn86] é balancear detalhes locais e o contexto global em relação ao nó selecionado (em foco) pelo usuário. Detalhes locais são necessários para dar aos usuários informações sobre suas possibilidades de navegação, dependentes da sua localização na estrutura do documento. O contexto global é importante para dar aos usuários informações sobre sua posição dentro da estrutura global do documento.

A estratégia básica do olho-de-peixe propõe uma função de grau de interesse que atribui a cada nó um valor representando seu interesse para o usuário. Esta função é decomposta em duas componentes. A primeira é chamada de importância a priori e descreve a importância intrínseca de um nó considerando a estrutura do documento. A segunda componente determina a distância entre o nó cuja função está sendo calculada e o nó em foco pelo usuário. A essência da visão olho-de-peixe é que a função de grau de interesse aumenta com a importância a priori (API) e diminui com a distância (D).

Em um modelo conceitual com composições aninhadas, o conceito de *perspectiva* é definido pela seqüência de composições, da mais externa para a mais interna, utilizada para alcançar um determinado nó. Na estratégia olho-de-peixe estendida, o grau de interesse do usuário para cada nó depende da perspectiva na qual está inserido. Um mesmo nó terá um valor da função de grau de interesse para cada uma de suas perspectivas. A função de grau de interesse deve ser calculada para todas as perspectivas da estrutura do documento. Detalhes sobre essa função podem ser encontrados em [Much96].

Já que um mesmo nó pode ser inserido em diferentes composições, ele também pode ser desenhado em diferentes posições no mapa. Por outro lado, exibir um nó em uma de suas perspectivas não significa que será exibido em todas as outras. Como a posição de um nó em uma dada perspectiva depende de quais perspectivas estão sendo exibidas no mapa, os nós de um documento não têm uma posição fixa na tela. Para reduzir esse problema, o usuário pode definir algumas perspectivas como pontos de referência, isto é, perspectivas que serão sempre exibidas no mapa, pelo menos parcialmente, ajudando o usuário a melhorar o seu senso de localização. Esses pontos de referência são chamados *landmarks* e são específicos a cada usuário. Um usuário não precisa concordar com o conjunto de landmarks de qualquer outro usuário, cada usuário possuindo o seu conjunto.

As idéias propostas neste artigo consideram o fato de que, em modelos com nós de composição aninhados, as composições estruturam o documento. Assim, é possível construir editores que exploram o recurso de "explodir" ou "implodir" nós de composição, mostrando ou escondendo seus componentes, dependendo do interesse atual do usuário. Esse recurso limita a quantidade de informação detalhada apresentada no mapa, melhorando sua legibilidade.

Para ilustrar melhor as técnicas de filtragem utilizadas pelo editor de estruturas, as Figuras 5 e 6 apresentam um exemplo de visões olho-de-peixe na implementação do browser de hiperbase do sistema HyperProp [SoCR95]. O documento apresentado é representado por um nó de composição *A* (toda a janela): uma pequena parte de um relatório sobre o Carnaval Brasileiro. As relações de inclusão em nós de composição são representadas pela inclusão de retângulos (nós) em outros retângulos (nós de composição), como usual. As figuras mostram como o mapa é modificado dependendo do nó em foco e do nível de detalhe escolhido pelo usuário.

Para apresentar as visões olho-de-peixe, foi utilizada uma técnica de layout automático de grafos [SuMi91]. Como a estrutura dos nós e elos visíveis muda muito, o diagrama pode mudar freqüentemente de uma visão para outra. Mudanças repentinas e drásticas de diagramas geralmente destroem o mapa mental do usuário, diminuem a eficiência do trabalho e podem causar desorientação. Com o objetivo de reduzir esses problemas, o editor mostra mudanças dos diagramas com animação. Esse recurso reduz a mudança visual instantânea, preservando o mapa mental do usuário e ajudando sua orientação.

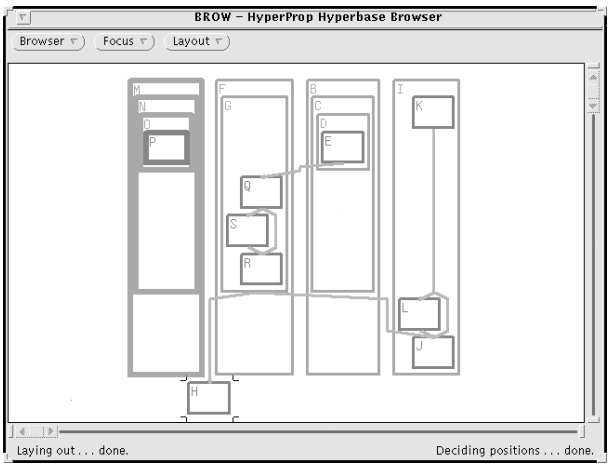
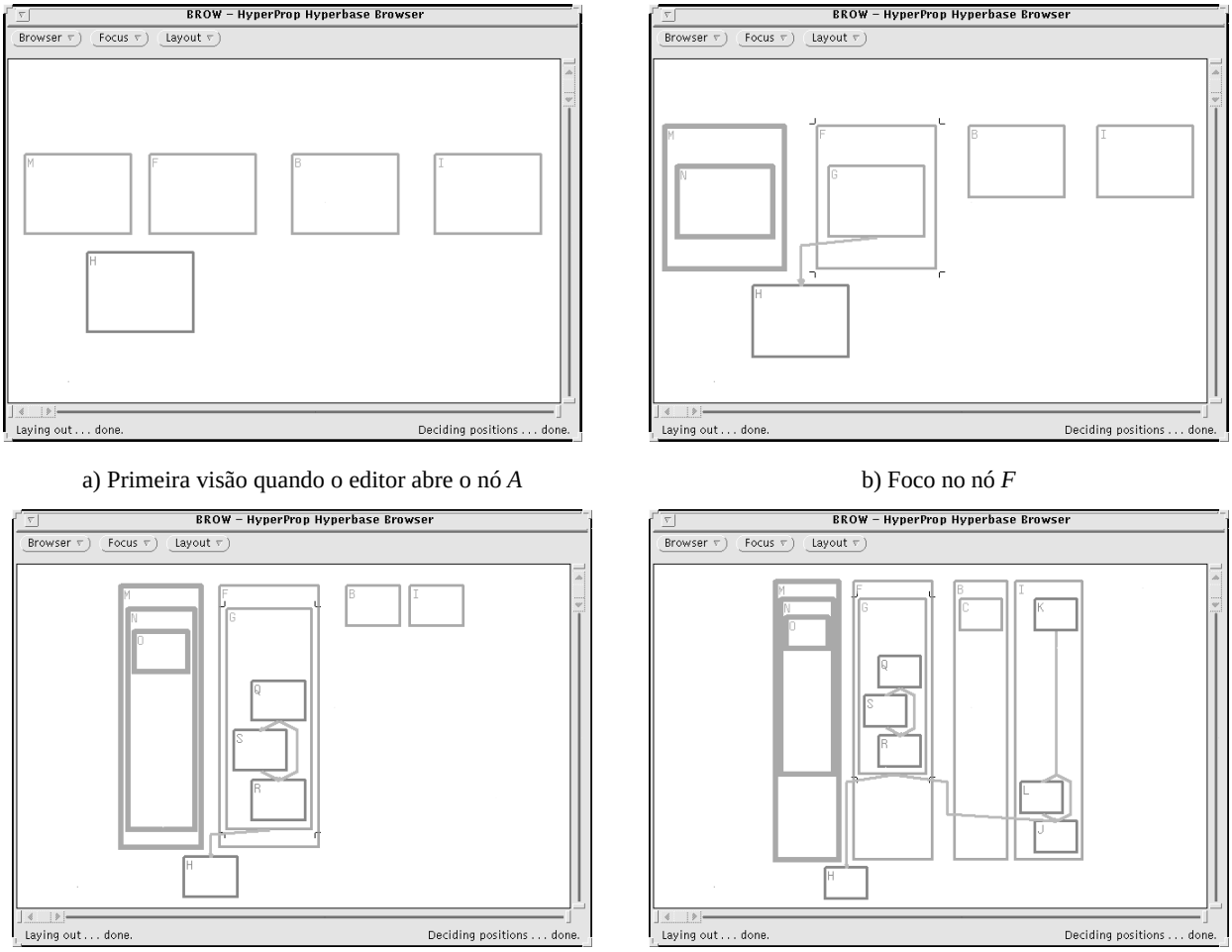


Figura 5 - Visão detalhada dos componentes aninhados do nó de composição *A*. *<M,N,O,P>* é um landmark.

A edição da estrutura de documentos tem sido investigada por vários projetistas de sistemas multimídia/hipermídia. Geralmente, os sistemas não oferecem uma ferramenta genérica que pode ser moldada para satisfazer aos interesses do usuário quando edita ou navega pela estrutura de nós e elos. Normalmente, ou o sistema oferece um mapa global mostrando uma visão ponto-a-ponto de todos os nós e elos de um documento, ou mostra um mapa local indicando a posição do usuário, de onde ele veio e para onde ele pode ir, apresentando apenas os nós que são adjacentes ao nó corrente.

Visões estruturais para documentos foram propostos pelos sistemas Neptune, Notecards, CYBERMAP, Storyspace, SHADOCS, Intermedia, entre outros. A abordagem mais interessante e que foi estendida no presente trabalho é a do CYBERMAP [Gloo91], que agrupa nós em estruturas chamadas HyperDrawers, através da análise de seu conteúdo e de suas características, sem considerar os elos do documento. A desvantagem dessa técnica é que as relações modeladas por elos são ignoradas, além da criação dos HyperDrawers ser bastante complexa. Ao contrário, no modelo com composições aninhadas utilizado neste trabalho, os grupos de nós são dados diretamente pela hierarquia de composições e pelos elos. Esse fato permite a construção de visões que não exibem necessariamente todos os nós do documento de forma simples e sem perder a semântica dos elos.



a) Primeira visão quando o editor abre o nó *A*

b) Foco no nó *F*

c) Foco no nó *G*

d) Foco no nó *G* (mais detalhes)

Figura 6 - Exemplo das visões olho-de-peixe de um nó de composição *A*.

3.2 - Visões Temporal e Espacial

Sincronização se refere aos mecanismos utilizados no alinhamento de eventos no domínio do tempo e do espaço, onde um evento é definido pela exibição ou seleção de uma região de um documento. Um editor de sincronismo deve permitir ao autor especificar os eventos relevantes e definir as relações entre eles. Através de uma interface gráfica o editor de sincronismo permite a definição das posições temporais e espaciais de um objeto, em relação a um ponto específico no tempo ou a outros objetos, e também a definição de mudanças do comportamento na apresentação de um nó. Essas relações são capturadas pelas entidades elo e descritor, respectivamente, anteriormente descritas.

O principal objetivo do editor de sincronismo é facilitar a edição dos aspectos de apresentação de documentos multimídia. Com esse propósito, a interface gráfica mostra duas visões dos documentos: a visão temporal (time view) e a visão espacial (spatial view).

Na time view, ilustrada pela Figura 4, os objetos são desenhados nas áreas de áudio e display, que correspondem às abstrações dos respectivos dispositivos da plataforma de exibição. Os nós são representados por retângulos, cujos comprimentos correspondem às durações esperadas de suas exibições. Nessa visão, os elos, cujas extremidades são eventos previsíveis, isto é, eventos cujos momentos de ocorrência podem ser calculados a priori, são desenhados como linhas que conectam os nós. Esse tipo de elo é chamado *elo de sincronismo*.

A visão é focada em um objeto chamado *objeto base*, que é posicionado na origem do eixo do tempo. Os demais objetos ficam posicionados, no tempo especificado, em relação ao objeto base, que corresponde ao objeto em foco na visão estrutural, anteriormente descrita. As regras para inclusão de objetos na time view

são as seguintes:

- *Inclusão de nós antecessores*: Se um nó incluído na visão for destino de um e apenas um elo de sincronismo (relação 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 sincronismo que têm a condição de disparo satisfeita, isto é, os nós em que é possível determinar sua exibição no tempo, são também incluídos na visão.

Note que essas regras são recursivas e são aplicadas a todos os nós incluídos na *time view*, iniciando-se pelo objeto base. A coleção de nós resultante da aplicação dessas regras de inclusão é chamada de *time chain*.

O usuário pode incluir explicitamente um nó na time view, para a definição de uma nova relação de sincronismo, simplesmente o arrastando ("dragging") da visão estrutural para a visão temporal. Quando um nó *N* é incluído pelo usuário na time view, ele é inicialmente posicionado na origem do eixo do tempo. Um elo de sincronismo (relação 1:1), tendo o objeto base como origem e *N* como destino, é, então, automaticamente criado pelo editor.

O usuário pode trocar a posição relativa de objetos, desde que seja mantida a consistência do sincronismo, verificada pelo editor. Quando o usuário destrói um elo *E* na time view, os nós e elos que foram posicionados iniciando em *E* deixam de ser mostrados na time view, isto porque a *time chain* desde o objeto base foi também destruída.

O usuário também pode definir o sincronismo entre objetos através de relações de alto nível: *exiba ao término de, exiba iniciando ao mesmo tempo, exiba iniciando e terminando ao mesmo tempo*. Essas relações provocam a criação automática de elos entre os nós envolvidos.

A time view possui um mecanismo de *scroll* que permite a visualização de todos os instantes de tempo de uma *time chain*. Ela possui ainda uma barra de tempo móvel (mostrada na Figura 4) que pode ser posicionada em qualquer instante. Se a barra de tempo estiver sobre um ou mais nós, esses nós serão mostrados na spatial view, dentro dos canais onde eles serão apresentados.

Na *spatial view* (ver Figura 4), o usuário edita o sincronismo espacial dos objetos posicionando-os em canais apropriados. Através desse posicionamento, o usuário define, por exemplo, o espaço no display que será usado para mostrar uma janela com uma imagem gráfica, o volume que será usado para tocar um nó de áudio, etc. Quando o usuário muda o comportamento (posição, volume, etc.) de um objeto na spatial view, o sistema inclui, na lista de operações do descritor, uma operação que descreve a mudança de comportamento do nó.

Se, na spatial view, existirem dois ou mais objetos de áudio exibidos no mesmo canal ao mesmo tempo, o tamanho dos retângulos indica como a mixagem dos sinais será feita. Analogamente, a ordem de sobreposição dos retângulos no canal de display determina como as janelas serão apresentadas no monitor de vídeo. A referência [CoSS96] traz uma descrição completa da interface com o usuário do editor de sincronismo.

Os três principais paradigmas de edição da apresentação de documentos multimídia são: *timeline*, *scripting* e *baseado em eventos*. No paradigma de timeline, a sincronização temporal de cada componente é especificada pelo posicionamento manual no eixo do tempo [HoSA89]. A desvantagem desse paradigma é que as edições em um componente pode requerer o reposicionamento manual dos demais componentes. Também é impossível nesse paradigma o posicionamento relativo de objetos de duração indefinida e a representação de interações com o usuário. Outro paradigma para definir a sincronização usa linguagens de script [FiTD87], onde um programa é construído contendo a descrição da apresentação. Apresentações pequenas podem ser facilmente construídas, mas a especificação de apresentações de grande porte torna-se uma tarefa difícil. O último paradigma se baseia na especificação do sincronismo através de relações entre eventos definidos nos componentes do documento. No editor aqui apresentado, esses relacionamentos são modelados pelos elos, como no MHEG e no HyTime. Outros sistemas que usam o paradigma baseado em eventos são o Firefly [BuZe92] e o CMIFed [RJMB93].

4 - Conclusões

Este artigo discutiu as características de um editor para estrutura e sincronização de documentos multimídia através de suas visões estrutural, temporal e espacial.

Além da edição gráfica integrada da estrutura e sincronismo de um documento multimídia, várias outras contribuições permeiam o trabalho. Na área de grafos, novos algoritmos para a implementação da visão olho-de-peixe em grafos compostos são utilizados no editor de estrutura apresentado [Much96]. Este editor, a princípio implementado para o sistema HyperProp, hoje se encontra em utilização no sistema

para modelagem de software orientado a objetos 2GOOD e no ambiente DisCo para programação distribuída baseado no paradigma de configuração. A sincronização espacial e temporal do editor de sincronismo se baseia em um modelo conceitual de apresentação, cujo paradigma baseado em eventos [SoSC95] permite, entre outras facilidades, a verificação formal da consistência do sincronismo de um documento, bem como a criação de um estrutura de dados adequada a um escalonador de eventos, em desenvolvimento.

A ferramenta apresentada foi implementada sobre o sistema hipermídia HyperProp [SoCR95]. A implementação foi feita sobre a plataforma UNIX usando a linguagem C++. Para implementar a visão estrutural, foram utilizadas algumas rotinas do editor de grafos compostos D-ABDUCTOR [Misu94], da FUJITSU Laboratories LTD, que fornece o layout automático de grafos e os recursos de animação. Como essas rotinas eram dependentes da interface utilizada pelo sistema D-ABDUCTOR, foi adotado o mesmo pacote gráfico para a construção da janela da visão estrutural, o X-View. As visões temporal e espacial foram implementadas utilizando um conjunto de ferramentas portáteis para a construção de interfaces chamados IUP/LED e CD. A utilização de ferramentas distintas para a construção das interfaces foi considerada, inicialmente, irrelevante para o trabalho. Um dos próximos passos é homogeneizar a interface do editor gráfico integrado, utilizando uma ferramenta portátil.

REFERÊNCIAS

[BuZe92] Buchanan, M.C.; Zellweger, P.T.; Specifying Temporal Behavior in Hypermedia Documents, *Proceedings of European Conference on Hypertext*, ECHT'92. Milano. Dezembro 1992.

[CoSS96] Costa, F.; Souza, G.; Soares, L.F.; Editor Gráfico para Sincronização Temporal e Espacial de Objetos Multimídia/Hipermídia, *Anais do II Workshop de Sist. Hipermídia Distribuídos*, Fortaleza, Maio, 1996

[FiTD87] Fiume,E.; Tsichritzis, D.; Dami, L.; A Temporal Scripting Language for Object-oriented Animation. *Proc. Eurographics'87*. Amsterdam. 1987.

[Furn86] Furnas, G.; Generalized Fisheye Views, *Proceedings of CHI'86 Human Factors in Computing Systems*, Boston, Abril, pp. 16-23, 1986

[Gloo91] Gloor, P.; CYBERMAP Yet Another Way of Navigating in Hyperspace, *Proceedings of the Third ACM Conference on Hypertext*, pp. 107-121, San Antonio, Texas, Dezembro, 1991

[HaSc90] Halasz, F.G.; Schwartz, M.; The Dexter Hypertext Reference Model. *NIST Hypertext Standardization Workshop*. Gaithersburg. Janeiro, 1990.

[HoSA89] Hodges, M.E.; Sasnett, R.M.; Ackerman, M.S.; Athena Muse: a Construction Set for Multimedia Applications. *IEEE Software*. Janeiro, 1989.

[MHEG95] Multimedia and Hypermedia Information Coding Expert Group (MHEG). ISO/IEC DIS 13522-1 - Coded Representation of Multimedia and Hypermedia Information (MHEG), Part I: MHEG Object Representation, Base Notation (ASN.1), Setembro, 1995

[MiSu91] Misue, K.; Sugiyama, K.; Multi-viewpoint Display Methods: Formulation and Application to Compound Graphs. *Proceedings of the Fourth Conference on Human-Computer Interaction*, Stuttgart, F.R. Germany, Setembro 1-6, 1991

[Misu94] Misue K.; D-ABDUCTOR 2.30 User Manual, Institute for Social Information Science. *FUJITSU LABORATORIES LTD.*, Japan, 1994

[Much96] Muchaluat, D.C.; Editores e Trilhas para Documentos Hipermídia Baseados em Modelos com Composições Aninhadas. *Tese de Mestrado, Departamento de Informática, PUC - Rio*, Brasil, Março, 1996

[PuGu90] Puttress, J.J.; Guimarães, N.M.; The Toolkit Approach to Hypermedia. *Procceedings of European Conference on Hypertex*, ECHT'90. 1990.

[RJMB93] van Rossum, G.; Jansen, J.; Mullender, K. S.; Bullterman, D.; CMIFed: A Presentation Environment for Portable Hypermedia Documents. *Proceedings of ACM Multimedia'93*. Anaheim, CA, EUA. 1993.

[SoCC93] Soares, L.F.G.; Casanova, M.A.; Colcher, S.; An Architecture for Hypermedia Systems Using MHEG Standard Objects Interchange. *Proceedings of the Workshop on Hypermedia and Hypertext Standards*. Amsterdam, The Netherlands. Abril 1993.

[SoCR95] Soares, L.F.; Casanova, M.A.; Rodriguez, N.L.R.; Nested Composite Nodes and Version Control in an Open Hypermedia System. *Information Systems*, Special issue on Multimedia Information Systems, Vol. 20, n. 6, pp.501-519, 1995

[SoSC95] Souza; G.; Soares, L.F.; Casanova, M.A.; Synchronization Aspects of a Hypermedia Presentation Model with Composite Nodes. *ACM Workshop on Effective Abstractions in Multimedia*, in connection with ACM Multimedia'95, San Francisco, EUA. Novembro 1995.