Delineamento da Arquitetura de um Sistema para Processamento de Hiperdocumentos Multimídia

(Resumo)

Luiz F.G. Soares[1], Noemi Rodriguez[1], José L. Rangel[1],
Maria Júlia D. Lima[1], Luiz Tucherman[2], Marco A. Casanova[2]

[1]Departamento de Informática
Pontifícia Universidade Católica do RJ
R. Marquês de S. Vicente, 225
22.453, Rio de Janeiro, RJ - Brasil

[2]Centro Científico Rio
IBM Brasil
P.O. Box 4624
20.001, Rio de Janeiro, RJ - Brasil

Este trabalho resume brevemente o modelo conceitual e a arquitetura de um sistema localmente distribuído para processamento de documentos ora em desenvolvimento. As aplicações previstas para o sistema são aquelas típicas do tratamento de documentos em um ambiente de escritório, tais como tele-conferência.

O modelo conceitual proposto possui dois conceitos básicos, nó e elo, como usual, distinguindo porém dois tipos diferentes de nós: terminais e de contexto. Intuitivamente, um *nó terminal* armazena dados que serão apresentados em algum meio de saída. Um nó de *contexto* é um tipo especial de nó cujo conteúdo consiste de um conjunto de nós (terminais e de contexto) e de um conjunto de elos entre estes nós. Um *elo* conecta dois nós, chamados de *extremidades* do elo, e indica, para cada extremidade, a região onde o elo está "ancorado". Por exemplo, para nós de texto, a região será simplesmente uma cadeia de caracteres dentro do texto e, para nós de imagem bidimensional, será um retângulo determinado por um par de coordenadas.

O conceito de nó de contexto permite organizar, hierarquicamente ou não, conjuntos de nós e oferece um mecanismo para definir visões diferentes do mesmo documento, sintonizadas para aplicações diferentes ou para classes distintas de usuários. Por exemplo, para organizar hierarquicamente um livro-texto sobre aspectos formais de Ciência da Computação, podemos definir inicialmente um nó de contexto B que contém um conjunto de nós representando os capítulos do livro e um conjunto de elos indicando a organização dos capítulos, que não precisa ser uma seqüência. De forma similar, para o i-ésimo capítulo, podemos definir um nó de contexto C_i que contém um conjunto de nós representando as seções do capítulo e um conjunto de elos indicando a organização das seções, e assim por diante. Além disto, como ocorre frequentemente, se uma organização diferente para os capítulos é mais adequada a uma segunda classe de leitores, podemos definir um outro nó de contexto B' contendo os mesmos nós que B (ou seja, os mesmos capítulos), mas um conjunto diferente de elos indicando a nova organização.

O sistema está estruturado em duas camadas principais. A *camada de nós* oferece armazenamento persistente de nós e elos, organizados nas seguintes classes principais. A classe *TOPO* define as características comuns a todas as classes na hierarquia da camada de nós. A classe *NÓ* especializa *TOPO* e descreve as características comuns a todos os tipos de nós, além daquelas herdadas de *TOPO*. A classe *NÓ* possui *NÓ_CONTEXTO* como subclasse, para capturar nós de contexto, e uma subclasse para cada mídia diferente que o sistema suporta. Finalmente, a classe *ELO* captura o conceito de elo.

A *camada de apresentação* implementa as facilidades navegacionais e oferece, para cada tipo de nó, representações alternativas, sintonizadas para classes diferentes de aplicações. Uma *representação* para a classe *NÓ* é uma outra classe *C* contendo um *método de conversão r*, que mapeia os objetos de *NÓ* em objetos de *C*, e um *método de reconversão* r^{-1} que mapeia os objetos de *C* em objetos de *NÓ*. A noção de representação estende-se também às subclasses de *NODE*.

Por exemplo, podemos definir a classe *ED* cujos métodos incluem os comandos de um certo editor ED e duas operações que mapeiam nós de texto em objetos de *ED* e vice-versa. Os atributos da classe *ED* incluem *ReprInterna*, cujos valores corresponderão ao formato interno do editor ED para texto. A classe *ED* será então uma representação para a classe *NÓ_TEXTO*. Se uma aplicação quiser manipular um nó de texto *N* através do editor ED, deverá solicitar à camada de apresentação que crie uma *ED*-representação *E* de *N* e, a partir daí, passará a utilizar *E*.

Em qualquer instante, uma aplicação pode ter acesso a várias representações e pode solicitar a criação de representações novas ou a eliminação de representações antigas, como resultado da navegação pelos nós e elos. A camada de apresentação associa então um histórico *h* a cada aplicação; o último elemento de *h* é o *estado corrente* da aplicação.

Finalmente, identificamos quatro maneiras de navegação: travessia de um elo (*navegação por elos*); seleção do conteúdo de um nó contexto (*navegação pela hierarquia de contextos*); solicitação de estados anteriores (*navegação pelo histórico*); e seleção de um nó através da especificação de uma consulta (*navegação lógica*).

Isto conclui o resumo do modelo conceitual e da arquitetura do sistema para processamento de hiperdocumentos multimídia. Planejamos trabalhar em um futuro imediato na formalização destes pontos e no detalhamento dos vários subsistemas, como controle de versões e autorização. A implementação em si prosseguirá gradualmente de um sistema mono-usuário para o sistema distribuído multi-usuário completo. A especificação das aplicações dar-se-á em paralelo.