Âncoras e Elos em Nós de Composição*

Luiz Fernando G. Soares
Depto. de Informática, PUC-Rio
R. Marquês de São Vicente 225
22453-900 - Rio de Janeiro, Brasil
lfgs@inf.puc-rio.br

Marco Antônio Casanova
Centro Científico Rio, IBM Brasil
Av. Pres. Vargas 824
20071-001 - Rio de Janeiro, Brasil
casanova@vnet.ibm.com

Guido L. de Souza Filho
DIMAp, UFRN
Campus Universitário, Lagoa Nova
59072-970- Natal-RN, Brasil
guido@dimap.ufrn.br

Resumo

Este artigo discute a definição de âncoras e elos em nós de composição aninhados, isto é, em nós de composição que contêm outros nós de composição. São consideradas questões como herança de elos e referências a nós, que tornam-se bastante complexas na presença de nós de composição aninhados. A discussão é então estendida para incluir âncoras e elos virtuais, definidos por expressões em uma linguagem apropriada. Por fim, são endereçadas questões relacionadas a âncoras e elos na presença de versões de objetos, como por exemplo, a manutenção automática de uma referência à última versão de um nó.

Abstract

This paper focuses on the definition of anchors and links for nested composite nodes, that is, composite nodes containing other composite nodes. It considers questions such as link inheritance and node reference, which become fairly complex in the presence of nested composite nodes. Then, the paper extends the discussion to include virtual anchors and links, defined by expressions in an appropriate language. Finally, it addresses some issues related to anchors and links in the presence of object versioning, such as how to maintain automatic reference to the last version of a node.

1 - Introdução

A definição de âncoras e elos em nós de composição dá origem a problemas que precisam ser cuidadosamente analisados na definição de um modelo conceitual para sistemas hipermídia. A complexidade aumenta ainda mais se o modelo: (1) inclui nós de composição aninhados; (2) permite que um mesmo nó possa estar contido em nós de composição diferentes; (3) inclui objetos virtuais, isto é, objetos definidos por expressões notadas em uma linguagem apropriada; e (4) fornece facilidades para o gerenciamento de versões de objetos.

Na verdade, os nós de composição aninhados isoladamente já incitam questões interessantes, por exemplo: Como referenciar nós localizados dentro dos nós de composição? Como definir elos entre nós de composição aninhados? Pode um elo ser herdado pelos nós aninhados dentro de um nó de composição? Como definir herança de

* O trabalho apresentado neste artigo foi parcialmente financiado pelo projeto HyperProp-ProTeM do CNPq-Brasil.

elo? Objetos virtuais e o tratamento de versões de objetos levantam outras questões, por exemplo: Como definir elos e âncoras virtuais? Como referenciar uma versão particular de um nó? Como manter automaticamente uma referência para a última versão de um nó?

Este artigo enfoca a definição de âncoras e elos em nós de composição aninhados, respondendo questões como as levantadas no parágrafo anterior. Também são tratadas as definições de âncoras e elos virtuais, e são endereçadas questões relacionadas a âncoras e elos na presença de versões de objetos.

A definição de nó de composição deste artigo segue a da referência [SoCR95] que é a adotada no sistema HyperProp [SoCC93]. Esta definição generaliza a de nós de contexto introduzida no sistema Neptune [DeSc85, DeSc87], a qual na realidade baseou-se em algumas idéias do PIE [GoBo87], dos Intermedia webs [Meyr85], dos fileboxes e browsers do Notecard [Hala88], e dos Tree Items do KMS [AkCY88]. Igualmente, os contextos do HyperPro [Oste92] e os composite objects do HyperBase [ScSt90] são muito semelhantes aos nós de composição do HyperProp. Nossa definição de âncoras provê as mesmas facilidades que as do Intermedia e do Neptune (HAM), permitindo que as âncoras apontem para regiões internas aos nós origem e destino dos elos. Por outro lado, ela difere das do KMS, do NoteCard e do HyperCard, onde a âncora que é destino de um elo deve ser um nó inteiro. Nós modelamos as âncoras como atributos do nó, ao contrário do HAM, que trata as âncoras como atributos do elo. Por esse motivo, modificações no conteúdo de um nó não exigem modificações nos elos em nosso modelo, o que não acontece no HAM. As referências [DeSc87, Oste92, Haak92, SoCR94] endereçam especificamente o tratamento de versões em sistemas hipermídia. Em especial, o HyperProp trata grupos de versões como nós de composição, como fazem também o HyperPro e o CoVer.

Este artigo está assim organizado. A Seção 2 apresenta sucintamente um modelo conceitual genérico com nós de composição e define âncoras e elos no contexto desse modelo. A Seção 3 trata dos elos e âncoras virtuais e endereça o controle de versões. A Seção 4 é reservada para as conclusões.

2 - Um Modelo Conceitual com Nós de Composição

2.1 Nós

A Figura 1 ilustra a hierarquia de classes de um modelo conceitual básico para documentos hipermídia que suporta composições.

Uma *entidade* é um objeto que possui como atributos[1] pelo menos um *identificador único*, uma *lista de controle de acesso* e um *descritor*. O identificador único possui o significado usual. Para cada atributo da entidade, a lista de controle de acesso possui uma entrada que associa um usuário[2] ou grupo de usuários aos seus respectivos direitos de

[1] O nome do atributo de uma entidade será empregado com frequência referindo-se ao valor do atributo. Quando o contexto não permitir essa simplificação, será usado explicitamente o termo *valor do atributo*.

acesso ao atributo. O descritor da entidade contém informações que determinam como a entidade deve ser apresentada ao usuário, desempenhando um papel semelhante ao de uma presentation specification no modelo Dexter [HaSc90].

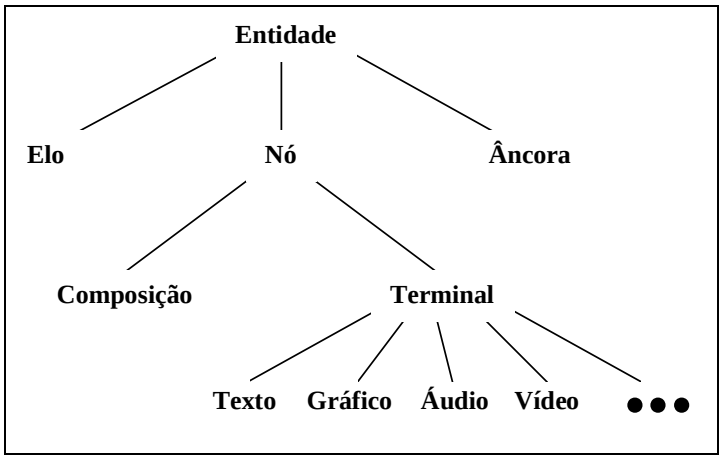

Figura 1: Hierarquia de Classes do Modelo Conceitual Básico.

Um *nó* é uma entidade que possui como atributos adicionais uma *lista de âncoras* e um *conteúdo*. O primeiro conceito será discutido em detalhes na Seção 2.2. A definição exata do atributo conteúdo de um nó depende da classe do nó.

Um *nó terminal* é um nó cujo conteúdo e lista de âncoras dependem da aplicação. O modelo permite que a classe nó terminal seja especializada em outras classes (*texto*, *áudio*, *vídeo* etc.).

Um nó *de composição C* é um nó cujo conteúdo é um conjunto *L* de nós e elos (as Seções 2.2 e 2.3 discutem restrições em nós de composição relacionadas a âncoras e elos). Uma entidade *E* que pertence a *L* é um *componente de C*, diz-se que *E está contida em C*. Adicionalmente, uma entidade *A* está *recursivamente contida em B* se e somente se *A* está contida em *B* ou *A* está contida em uma entidade recursivamente contida em *B*. A classe nó de composição pode ser especializada em outras classes, como pode ser verificado nas referências [Casa91, SoCR94, SoCR95].

O conteúdo de *C* pode ser definido como uma lista ordenada, como em [SoCR95], para que uma entidade possa ser incluída mais de uma vez em *L*. Essa alternativa é útil para a definição de mecanismos de navegação sofisticados, como as trilhas [SoCR95], mas requer definições mais elaboradas para os conceitos introduzidos nas seções seguintes.

Nós de composição permitem organizar, hierarquicamente, ou não, conjuntos de entidades, isto é, conjuntos de nós e elos. Por conseguinte, eles ajudam a estruturar os hiperdocumentos, minorando o problema denominado “lost in hyperspace” [Hala88, Hala91]. Por exemplo, considere um documento sobre Poesia Inglesa do Século XVI no qual trabalha uma equipe de teatrólogos. O documento pode ser modelado como um nó

---

[2] A palavra usuário no contexto deste artigo posui múltiplos significados: ela significa um usuário no sentido de uma pessoa, um processo de aplicação ou um programador de aplicação. Isto é, qualquer coisa ou qualquer pessoa que faça uso dos serviços definidos nas várias interfaces das camadas do sistema HyperProp.

de composição *E* que contém um nó de composição *S*, agrupando peças de Shakespeare, e outro nó de composição *C*, agrupando sonetos de Christopher Marlowe. O nó *S* pode conter, digamos, os nós texto *H* e *M* representando as peças "Hamlet" e "Macbeth" e *C* pode conter um nó texto *F* representando "Dr. Faustus". Esse exemplo será estendido na Seção 2.3 com a inclusão de elos.

A classe nó de composição pode ser especializada em outras classes, por exemplo, para organizar um grupo de versões do mesmo nó [Oste92, Haak92, SoCR94]. Uma aplicação pode então usar esses nós de composição para manter um histórico das versões de um documento, bem como para manter automaticamente uma referência para sua última versão, dois importantes problemas relacionados ao controle de versões [Oste92, Haak92, SoCR95]. Esse tópico será expandido na Seção 3.2.

Por fim, nós de composição podem ser usados para modelar a interação do usuário com um hiperdocumento, em conformidade com o paradigma de sessão de trabalho proposto pelo Modelo Dexter [HaSc90]. O leitor interessado deve consultar as referências [SoCR94, Haak92] para uma discussão detalhada sobre o uso de nós de composição para suportar trabalho cooperativo.

2.2 - Âncoras

Conforme já foi mencionado, um nó é uma entidade que possui entre seus atributos um conteúdo e uma lista de âncoras. Os elementos da lista de âncoras são chamados *âncoras* do nó. Uma âncora é uma entidade que possui como atributo adicional uma *região*. As âncoras de um nó são ordenadas em uma lista para facilitar o tratamento de suas versões, conforme será discutido na Seção 3.2. Note que uma âncora pode ser identificada: (1) por seu identificador único (uma vez que é uma entidade); ou (2) pelo identificador único do nó onde ela ocorre e sua posição na lista de âncoras desse nó. Note também que uma âncora pode pertencer a mais de uma lista de âncoras. A definição exata da região associada a uma âncora depende da classe do nó, como será discutido nos parágrafos que seguem.

Uma *região de um nó terminal* é definida como o símbolo especial λ, representando todo o conteúdo do nó, ou como um conjunto de *unidades de informação marcadas*. O conceito de unidades de informações marcadas é semelhante ao conceito de marked states introduzido pelo Firefly [BuZe92]. A noção exata do que constitui uma unidade de informação (UI) ou uma unidade de informação marcada é parte da definição do nó terminal. Por exemplo, uma UI de um nó vídeo pode ser um quadro, enquanto uma UI de um nó texto pode ser um caracter. Qualquer subconjunto das unidades de informação de um nó terminal pode ser marcado.

A lista de âncoras atua como interface externa de um nó terminal *N*, no sentido que uma entidade só pode ter acesso a regiões do conteúdo de *N* através de sua lista de âncoras. Por esse motivo, as âncoras isolam as outras entidades de modificações no conteúdo de *N*. Como um exemplo, considere um nó texto com uma âncora cuja região aponta para uma informação descrita no quinto parágrafo do texto. Assuma que um elo refere-se indiretamente a esta informação identificando apropriadamente essa âncora. Nesse caso, qualquer modificação no texto, por exemplo a inclusão ou exclusão de um parágrafo antes do quinto parágrafo do texto, é refletida apenas na região da âncora (que passa então a

apontar para o quarto ou sexto parágrafo do texto caso a operação tenha sido uma exclusão ou inclusão, respectivamente) e não afeta o elo.

Em um nó de composição *C*, a região associada a uma âncora pode ser o símbolo especial λ ou um subconjunto de *L* contendo apenas nós (λ novamente representa todo o conteúdo do nó). A lista de âncoras de *C* possui a mesma função que a de um nó terminal, isto é, agir como uma interface externa do nó, entretanto, nos nós de composição ela não isola as entidades de modificações no conteúdo de *C*. Em particular, os elos são potencialmente sensíveis a modificações no conteúdo de *C* (esse comentário será esclarecido imediatamente após a definição dos elos na Seção 2.3).

Para finalizar, cabe salientar que, ao contrário do modelo conceitual do WWW, a lista de âncoras foi aqui definida como um atributo do nó, separado do seu conteúdo. De fato, as âncoras podem ser definidas: (1) como parte do conteúdo de um nó; (2) como um atributo separado do conteúdo do nó; (3) como um atributo de outro objeto, por exemplo de um elo; (4) como um objeto inteiramente independente. Nos três últimos casos, ilustrados pelo HyperProp [SoCR94], pelo Microcosm [FHHD90] e pelo modelo apresentado em [Oste92], o usuário pode criar âncoras até mesmo em nós cujo conteúdo ele não pode modificar, como por exemplo um nó armazenado em um CD ROM.

2.3 - Elos

Um elo é uma entidade que possui três atributos adicionais: o *conjunto de extremidades de origem*, o *conjunto de extremidades de destino* e o *ponto de encontro*. Os valores dos dois primeiros atributos são conjuntos cujos elementos, chamados *extremidades* do elo, são pares da forma $<(N_k,...,N_1),A>$ onde N_1 é um nó, N_{i+1} é um nó de composição e N_i está contido em N_{i+1}, para todo $i \in [1,k)$, com k>0. *A* é uma âncora de N_1. O nó N_k é denominado nó base do elo. O atributo ponto de encontro define condições que devem ser satisfeitas nas extremidades de origem (por exemplo, extremidade foi selecionada ou apresentada) para que sejam executadas ações (por exemplo, exibir, suspender exibição, retomar exibição etc.) nas extremidades de destino do elo, conforme discutido nas referências [SoSo95, SoSC95].

O modelo também exige que, para todo elo *l* contido em um nó de composição *C*, para todas as extremidades $<(N_k,...,N_1),A>$ de *l*, o nó N_k seja o nó de composição *C* ou um nó contido em *C*.

Se um elo possuir uma única extremidade de origem $<(S_m,...,S_1),A>$ e uma única extremidade de destino $<(T_n,...,T_1),B>$, ele será denotado pelo par $(<(S_m,...,S_1),A>,<(T_n,...,T_1),B>)$ no texto que segue.

Os elos definem relacionamentos entre nós, como por exemplo relacionamentos *go to* (para referências) e relacionamentos de sincronização (como os definidos na proposta MHEG [MHEG93]). A possibilidade de relacionar *M* extremidades de origem com *N* extremidades de destino foi adotada no modelo com a intenção de fornecer suporte às aplicações onde, por exemplo, a seleção das *M* extremidades de origem de um elo deve implicar na exibição simultânea dos nós associados às suas *N* extremidades de destino. A definição das extremidades de um elo *l* como pares da forma $<(N_k,...,N_1),A>$ em alguns casos simplifica consideravelmente a modelagem de documentos. Contudo, se o conteúdo do nó de composição N_{i+1} for modificado com a remoção de N_i , o elo *l* terá que ser

modificado. Por esse motivo, a lista de âncoras de um nó de composição não isola inteiramente as outras entidades de modificações no conteúdo do nó, como no caso anteriormente descrito envolvendo N_{i+1} e *l*. Entretanto, se o elo possuir uma extremidade da forma $<(N_k),A>$, então a âncora *A*, nesse caso, isola as modificações em N_k.

A lista de nós em cada extremidade torna possível a definição de elos conectando informações que não estão contidas diretamente no mesmo nó de composição. Como um exemplo, considere novamente o nó de composição *E* que contém um nó de composição *S*, agrupando peças de Shakespeare, e outro nó de composição *C*, agrupando sonetos de Christopher Marlowe. Relembre também que o nó *S* contém os nós texto *H* e *M* representando as peças "Hamlet" e "Macbeth" e *C* contém um nó texto *F* representando "Dr. Faustus". Seria bastante razoável que a equipe de teatrólogos trabalhando com o documento desejasse registrar uma conexão entre "Hamlet" e "Dr. Faustus" (este elo poderia ser usado, por exemplo, para registrar uma conexão entre peças cujo tema principal é conflito). Esse elo pode ser definido em *E* como *(<(S,H),r>,<(C,F),s)>)*, onde as âncoras *r* e *s* permitem que a conexão seja feita com mais precisão, por exemplo, apontando as sentenças onde o conceito comum aparece primeiro.

Como outro exemplo, uma vez que *S* contém *H* e *M*, um elo conectando esses nós pode, a princípio, ser definido em *S* como l_S*=(<(H),i>,<(M),j>)*, onde *i* e *j* são âncoras válidas em *H* e *M*, como mostra a Figura 2.a. Uma alternativa a essa primeira opção seria criar um elo em *E* conectando *H* e *M*, esse elo seria definido como l_E*=(<(S,H),i>,<(S,M),j)>)*, como mostra a Figura 2.b. Note a diferença entre essas duas alternativas. O elo l_S será visto por todo documento que incluir *S* (o nó de composição agrupando as peças de Shakespeare provavelmente será compartilhado por diversos documentos), ao passo que o elo l_E só será visto em *S* pelos usuários que alcançarem *S* passando por *E* (veja a discussão sobre perspectiva na Seção 2.4).

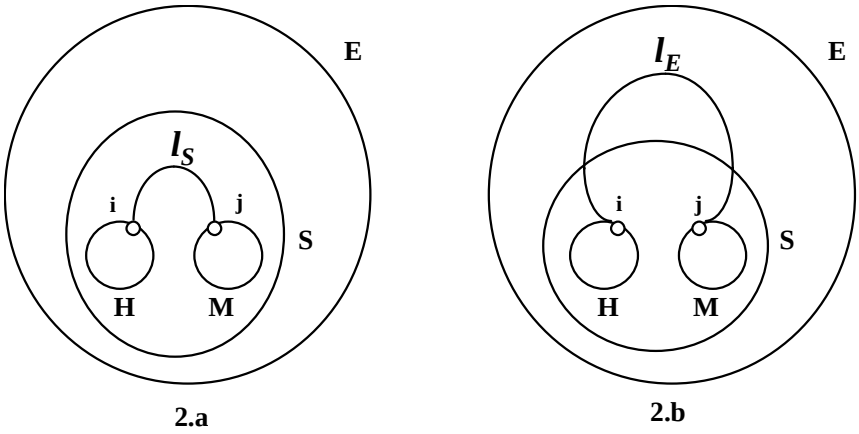

Figura 2: Elos em nós de composição.

O relacionamento entre *H* e *M* pode ser adicionalmente capturado por um conjunto de três elos, l_1, l_2 e l_3, como mostra a Figura 3. Nesse caso, o elo l_1 *=(<(S),m>,<(S),n>)* será definido em *E* e os elos l_2 *=(<(H),i>,<(S),m>)* e l_3 *=(<(S),n>,<(M),j>)* serão definidos em *S*. Note que, nesse caso, as âncoras isolarão as modificações nos nós de composição, ao contrário do exemplo mostrado na Figura 2.b.

Essas formas alternativas de capturar o relacionamento entre *H* e *M* abrem diferentes possibilidades para a apresentação desses nós dependendo de como eles são alcançados.

Por exemplo, em ambos os casos da Figura 2, assuma que o relacionamento definido pelo elo é "vá para a extremidade e apresente". Então, após a apresentação do nó *H*, se o usuário navegar através do elo, o nó *M* será apresentado. O usuário não saberá que *M* está contido em *S*. Por outro lado, na Figura 3, o relacionamento definido pelos elos l_3 e l_2 poderia ser "vá para a extremidade e apresente", ao passo que o relacionamento definido por l_1 poderia ser "vá para a extremidade apresente durante um minuto". Nesse caso o usuário ficaria sabendo que *M* está contido em *S* quando os elos l_2, l_1 e l_3 forem atravessados.

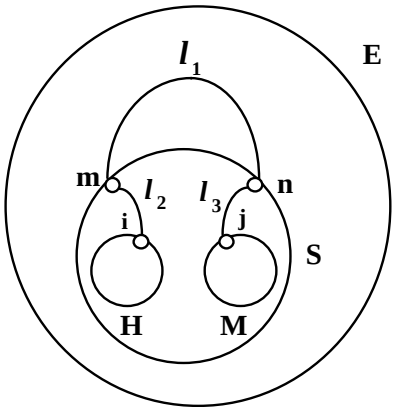

Figura 3: Forma alternativa de definir elos em nós de composição.

2.4 - Perspectivas e Elos Visíveis

Relembre que o modelo permite que nós de composição diferentes contenham o mesmo nó e que nós de composição sejam aninhados em qualquer profundidade. É necessário então introduzir o conceito de perspectiva para identificar através de qual sequência de nós de composição aninhados um dado nó está sendo observado, e a noção de elo visível para determinar quais os elos efetivamente tocam o nó segundo uma dada perspectiva.

A *perspectiva* de um nó *N* é uma sequência $\boldsymbol{P} = (N_m,...,N_1)$, com $m \geq 1$, onde $N_1=N$, o nó N_m não está contido em nenhum nó de composição, N_{i+1} é um nó de composição e N_i está contido em N_{i+1}, para $i \in [1,m)$. Uma vez que *N* é implicitamente determinado por $\boldsymbol{P}$, iremos nos referir a $\boldsymbol{P}$ simplesmente como uma perspectiva. Note que podem existir diferentes perspectivas para um mesmo nó *N*, se esse nó estiver contido em mais de um nó de composição. A *perspectiva corrente* de um nó em uma dada sessão do usuário é aquela atravessada pela última navegação para o nó na sessão.

Dado um nó N_1 e a perspectiva $\boldsymbol{P} = (N_m,...,N_1)$, diz-se que um elo *l* está *ativo para* N_1 *por* $\boldsymbol{P}$ se e somente se existe $r \in [1,m]$ tal que *l* está contido em N_r e *l* possui uma extremidade com uma das seguintes formas: $<(N_r,...,N_1,E_n,...,E_1),A>$ ou $<(N_{r-1},...,N_1,E_n,...,E_1),A>$, onde em ambos os casos o sufixo $E_n,...,E_1$ pode ser omitido. Além disso, se o sufixo não existir, diz-se que o elo está *integralmente ativo para* N_1 com a âncora *A visível em* N_1. Caso contrário diz-se que o elo está *parcialmente ativo para* N_1 com a âncora λ *visível em* N_1.

Note que o requisito na extremidade é consistente com a definição genérica de extremidade. Intuitivamente, o elo *l* deve possuir uma extremidade que contém N_1 e que segue a lista de nós de composição aninhados descendo até N_1.

Uma *multi-perspectiva* de uma lista de nós $N_1,...,N_k$ é uma lista $\boldsymbol{p}=(\boldsymbol{P}_1,...,\boldsymbol{P}_k)$ tal que $\boldsymbol{P}_i$ é uma perspectiva para N_i, para cada $i \in [1,k]$. Dada uma lista de nós $N_1,...,N_k$ e uma multi-perspectiva $\boldsymbol{p}=(\boldsymbol{P}_1,...,\boldsymbol{P}_k)$ para esses nós, diz-se que um elo *l* é *visível a partir de* $\boldsymbol{p}$ *por* $N_1,...,N_k$ se e somente se, para algum $r \in [1,k]$, o elo *l* está ativo para N_r por $\boldsymbol{P}_r$.

Por exemplo, considerando a Figura 4 assuma que: (i) o nó *A* contém os nós *B* e *Z*; (ii) o nó B contém os nós *C* e *D*; (iii) o nó *Z* contém *C* e *E;* e (iv) os nós de composição contêm os seguintes elos:

	Nó	***Nós contidos em***	***Elos contidos em***
(1)	*A*	*B* e *Z*	*(<(B,C),i>,<(Z,E),j>)*
(2)			*(<(B),m>,<(Z),n>)*
(3)	*B*	*C* e *D*	*(<(C),r>,<(D),k>)*
(4)	*Z*	*C* e *E*	*(<(E),s>,<(C),t>)*

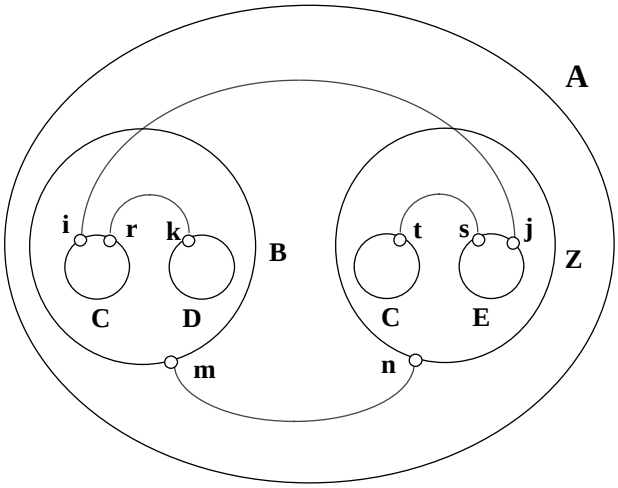

Figure 4: Exemplos de elos em nós de composição.

Nesse caso, são exemplos de multi-perspectivas e elos visíveis:

	Lista de Nós	***Multi-Perspectiva***	***Elos Visíveis***	***Explicação***
(5)	*C*	*((A,B,C))*	*(<(B,C),i>,<(Z,E),j>)*	de (1)
(6)			*(<(C),r>,<(D),k>)*	de (3)
(7)	*C*	*((A,Z,C))*	*(<(E),s>,<(C),t>)*	de (4)
(8)	*B*	*((A,B))*	*(<(B,C),i>,<(Z,E),j>)*	de (1)
(9)			*(<(B),m>,<(Z),n>*	de (2)
(10)	*C,C*	*((A,B,C),(A,Z,C))*	*(<(B,C),i>,<(Z,E),j>)*	de (1)
			(<(C),r>,<(D),k>)	de (3)
			(<(E),s>,<(C),t>)	de (4)

Note que o elo *(<(B,C),i>,<(Z,E),j>)* não é visível para *C* pela perspectiva *(A,Z,C)* pois *(B,C)* não é um sufixo dessa perspectiva.

Intuitivamente, a perspectiva corrente de um nó determina as âncoras que o usuário verá quando o nó for apresentado e os nós para os quais o usuário poderá navegar percorrendo um dos elos ativos quando selecionar uma das âncoras visíveis. Por exemplo, se o nó *C* for apresentado através da perspectiva *((A,B,C))* então, pelas linhas (5) e (6), as âncoras *i* e *r* serão visíveis em *C*. Se ele selecionar a âncora *i* então ele navegará para a âncora *j* de *E*, pela linha (5), e se ele selecionar a âncora *r* então ele navegará para a âncora *k* de *D*, pela linha (6). Da mesma maneira, pelas linhas (8) e (9), se o nó *B* for exibido pela perspectiva *((A,B))* então as âncoras visíveis em *B* serão o símbolo especial λ, indicando todo o nó *B*, e *m*, a partir de onde o usuário poderá navegar para a âncora *j* de *E* e para a âncora *n* de *Z*, respectivamente.

3 - Estendendo os Conceitos Básicos

3.1 - Âncoras e Elos Virtuais

Em termos gerais, uma *entidade virtual E* é uma entidade tal que o valor de pelo menos um atributo *A* é uma expressão ***e***, escrita em uma linguagem de consulta hipermídia formalmente definida, cuja avaliação resulta em um objeto do tipo apropriado. O atributo *A* é dito ser *virtual*. Quando alguma operação solicita o valor de *A*, o objeto resultante da avaliação de ***e*** é então retornado. A importância de entidades virtuais para o tratamento de versões é discutida na Seção 3.2.

Em particular, uma *âncora virtual* na lista de âncoras de um nó *N* é uma âncora cuja região é definida por uma expressão ***r***, que será computada quando um elo que possui uma extremidade tocando a âncora for atravessado. A avaliação da expressão ***r*** deve nesse caso resultar no valor de um segmento interno ao conteúdo de *N* ou no símbolo especial λ, se *N* for um nó terminal, ou em um subconjunto dos nós contidos em *N* ou no símbolo especial λ, se *N* for um nó de composição.

Um *elo virtual* é um elo *l* que possui no mínimo uma extremidade definida por uma expressão ***e***, que será computada quando o elo for atravessado. Se *l* estiver contido em um nó de composição C, então ***e*** deve retornar um nó base que é ou *C* ou um nó contido em *C*, conforme exigido na Seção 2.3. Os elos virtuais fornecem uma potente ferramenta de autoria, como a encontrada no Microcosm [FHHD90], que pode reduzir consideravelmente o esforço do autor ao compor hiperdocumentos.

Indo um passo adiante, define-se um *elo genérico l* para nós de classes $\boldsymbol{T}_1,...,\boldsymbol{T}_m$ como um elo cujo conjunto de extremidades de origem contém expressões $\boldsymbol{e}_1,...,\boldsymbol{e}_m$ tais que $\boldsymbol{e}_i$ retorna objetos que qualificam segmentos do conteúdo de nós da classe $\boldsymbol{T}_i$, para $i \in [1,m]$. O modelo considera que *l* conecta qualquer tupla de nós $M_1,...,M_m$ das classes $\boldsymbol{T}_1,...,\boldsymbol{T}_m$ que possuem âncoras cujas regiões $r_1,...,r_m$ definem segmentos que satisfazem as expressões $\boldsymbol{e}_1,...,\boldsymbol{e}_m$, respectivamente.

Intuitivamente, um elo é genérico quando seu uso não se restringe a apenas um documento, podendo ele ser empregado em toda uma classe de documentos. Por exemplo, pode ser definido um elo que possui como extremidade de destino um nó vídeo mostrando uma vista aérea da floresta Amazônica e como extremidade de origem a

expressão **floresta Amazônica**. Então, qualquer nó texto com uma âncora cuja região contenha a sentença *floresta Amazônica* pode ser automaticamente ligado ao nó vídeo.

Seja *N* um nó representado um dado documento. Não seria razoável tentar casar todos os elos genéricos com *N*, criando as âncoras apropriadas, se o número de elos genéricos for muito grande. Igualmente, poderia ser impraticável, se *N* for grande, tentar casar todas as regiões possíveis em *N* com a coleção de elos genéricos. No HyperProp, optou-se por uma solução para esse problema que permite ao usuário marcar uma área do documento e selecionar em um menu uma operação que tenta casar todas as regiões possíveis na área selecionada com a coleção de elos genéricos e mostra os casamentos bem sucedidos encontrados. Então, para cada região encontrada, se o usuário desejar, ele pode criar uma âncora *A* associada a região e um elo possuindo a mesma definição que o elo genérico avaliado, com a diferença que o elo criado possui *A* como âncora de uma extremidade de origem. O elo criado pertenceria a um nó de composição contendo todos os seus nós base, como exigido pelo modelo.

3.2 - Âncoras e Elos no Controle de Versões

Visando endereçar o problema de manter a história de um documento, o modelo conceitual introduzido na Seção 2 é estendido com a inclusão da classe grupo de versões.

Um *grupo de versões V* é um nó de composição tal que todos os nós contidos em *V* possuem listas de âncoras com o mesmo comprimento (a razão para essa restrição é explicada após a redefinição das extremidades de um elo que será feita mais adiante). Intuitivamente, *V* agrupa nós que representam versões da mesma entidade, em algum nível de abstração, sem que isso implique necessariamente que uma versão foi derivada a partir de outra. Os nós em *V* são chamados *versões correlatas*, e não precisam pertencer a mesma classe de nó. Note que um grupo de versões pode conter nós de composição, o que fornece um mecanismo explícito para tratar versões da estrutura de um documento.

O relacionamento de derivação é explicitamente capturado pelos elos em *V*. Diz-se que v_2 foi *derivada a partir de* v_1, se existir um elo da forma $(<(v_1),i_1>,<(v_2),i_2>)$ em *V*. As âncoras nesse caso simplesmente permitem uma maior precisão sobre que parte de v_1 gerou que parte de v_2. Não são impostas restrições ao relacionamento de derivação, exceto que ele deve ser acíclico.

Um usuário pode adicionar manualmente nós (indicando explicitamente que eles são versões do mesmo objeto) e elos (para indicar explicitamente como as versões foram derivadas) a um grupo de versões, ou ele pode criar um nó a partir de outro invocando uma operação do controle de versões, que atualizará automaticamente o grupo de versões apropriado.

Uma aplicação possui várias opções para definir o nó de um grupo de versões *V* que ela considera ser a *versão corrente* de *V*, de acordo com um critério específico. Por exemplo, a aplicação pode reservar uma âncora *A* de *V* para manter uma referência para a versão corrente (da aplicação). De fato, se o conceito de âncora virtual definido na Seção 3.1 é suportado, a aplicação pode definir a região de *A* como uma expressão ***e*** que quando avaliada sempre resulta na versão corrente. Uma expressão ***e*** pode resultar em várias versões (por exemplo, “todas as versões criadas por João”), que podem ser interpretadas como versões alternativas e são retornadas na forma de um nó de composição. Como outro exemplo, a expressão ***e*** pode ser definida em um elo ou mesmo de uma forma mais

geral, como nos modelos descritos em [SoCR94, Oste92], onde a versão corrente é definida em um atributo do nó de composição, para todos os seus nós componentes.

Na Seção 2, definiu-se uma extremidade de um elo contido em um nó de composição *C* como um par da forma $<(N_k,...,N_1),A>$ tal que N_k deve ser *C* ou um nó contido em *C*, N_{i+1} é um nó de composição e N_i está contido em N_{i+1}, para todo $i \in [1,k)$, com k>0, e *A* é uma âncora de N_1. Contudo, para incorporar a noção de versão em um grupo de versões, é necessário redefinir uma extremidade de um elo contido em um nó de composição *C* como uma tripla $<(N_k,...,N_1),A,q>$ tal que:

- Para todo $i \in [1,k)$, N_{i+1} é um nó de composição, N_i está contido em N_{i+1} e N_k é ou *C* ou um nó contido em *C*.
- Se N_1 é um grupo de versões, então *A* é uma âncora de N_1 e $q \in [1,n)$, onde *n* é o comprimento comum das listas de âncoras dos nós em N_1. Nesse caso, *q* identifica uma âncora de cada nó em N_1, em especial no nó(s) especificado por *A*. E, diz-se que N_2 não contém N_1, mas sim o nó(s) especificado pela âncora *A*.
- Se N_1 é um nó terminal ou um nó de composição (que não é um grupo de versões), então *A* é uma âncora de N_1 e *q* é nulo.

Essa definição é consistente porque o modelo exige que todas as versões pertencentes a um grupo de versões possuam listas de âncoras com o mesmo comprimento e que as âncoras de um nó estejam ordenadas. Assim, é possível referenciar a i-ésima âncora de uma versão em um grupo de versões sem necessariamente especificar a versão.

4 - Conclusões

Este artigo endereçou primeiramente os problemas encontrados na definição de âncoras e elos em nós de composição aninhados. O artigo também abordou os problemas adicionais incitados pela introdução de âncoras e elos virtuais e pela presença de suporte para tratamento de versões.

Entre outros pontos, foi equacionado o problema do isolamento de modificações no conteúdo de um nó tratando as âncoras como interfaces do nó. Foi explorado o aninhamento de nós de composição com a introdução de um formato bastante genérico para os elos, permitindo definir suas extremidades como sequências de nós de composição aninhados terminando em uma âncora. A herança de elos foi então tratada com a noção de perspectiva. Fazendo uso de âncoras e elos virtuais, apontou-se uma solução para o problema da manutenção automática de uma referência para a última versão de um nó.

As soluções aqui propostas foram discutidas com base na versão corrente do Modelo de Contextos Aninhados [Casa91, SoCR94, SoCR95] adotado e implementado no sistema HyperProp [SoCC93]. Em relação às anteriores, a versão aqui discutida acrescenta pequenos ajustes nos conceitos de perspectiva, lista de âncoras, elos e introduz o conceito de entidade virtual. No sistema HyperProp foram implementados todos os conceitos discutidos neste artigo excetuando-se o de entidade virtual que, contudo, está sendo abordado em um trabalho em andamento.

O trabalho apresentado neste artigo pode ser estendido em várias direções tais como: a especificação de uma linguagem para definir expressões nas entidades virtuais, ou a exploração das condições e ações associadas aos elos, tópico já endereçado em [SoSo95, SoSC95] que precisa, entretanto, ser mais aprofundado.

Referências

[AkCY88] Aksscyn, R.M.; McCracken, D.L.; Yoder, E.A. "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations". *Communications of ACM*, Vol.31, No. 7. June 1988.

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

[Casa91] Casanova, M.A.; Tucherman, L.; Lima, M.J.; Rangel Netto, J.L. Rodriguez, N.R.; Soares, L.F.G. "The Nested Context Model for Hyperdocuments". *Proceedings of Hypertext ' 91*. Texas. December 1991.

[DeSc85] Delisle, N.; Schwartz, M. "Neptune: A Hypertext System for CAD Applications". *Proceedings of ACM SIGMOD '85*. Washington, D.C. May 1985.

[DeSc87] Delisle, N.; Schwartz, M. "Context - A Partitioning Concept for Hypertext". *Proc. of Computer Supporteed Cooperative Work*. December 1987.

[FHHD90] Fountain, A.; Hall, W.; Heath, I.; Davis, H. "Microcosm: An Open Model for Hypermedia wiyh Dynamic Linking". *Proceedings of European Conference on Hypertext, ECHT'90*. Cambridge University Press. 1990, pp. 298 -311.

[GoBo87] Goldstein, I.; Bobrow, D. "A Layered Approach to Software Design". *Interactive Programming Environments*. McGraw Hill,p.387-413. New York. 1987.

[Haak92] Haake, A. "Cover: A Contextual Version Server for Hypertext Applications". *Proceedings of European Conference on Hypertext, ECHT'92*. Milano. December 1992.

[Hala88] Halasz, F.G. "Reflexions on Notecards: Seven Issues for the Next Generation of Hypermedia Systems". *Comm. of ACM*, Vol.31, N. 7. July 1988.

[Hala91] Halasz, F.G. "Seven Issues Revisited". *Final Keynote Talk at the 3rd ACM Conference on Hypertext*. San Antonio, Texas. December 1991.

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

[Meyr85] Meyrowitz, N. "Intermedia: The Architecture and Construction of an Object-Oriented Hypermedia System and Applications Framework". *Proceedings of the Conference on Object-Oriented Programming Systems, Languages and Applications*. Portland, Oregon. September 1985.

[MHEG93] MHEG. "Information Technology - Coded Representation of Multimedia and Hypermedia Information Objects - Part1: Base Notation. *Committee Draft ISO/IEC CD 13522-1*. July 1993.

[Oste92] Osterbye, K. "Structural and Cognitive Problems in Providing Version Control for Hypertext". *Proceedings of European Conference on Hypertext, ECHT'92*. Milano. December 1992.

[ScSt90] Schütt, H.A.; Streitz, N.A. "HyperBase: A Hypermedia Engine Based on a Relational Database Management System". *Procceedings of European Conference on Hypertex, ECHT'90. 1990.*

[SoCC93] Soares, L.F.G.; Casanova, M.A.; Colcher, S. "An Architecture for Hypermedia Systems Using MHEG Standard Objects Interchange". *Information Services & Use*, vol.13, no.2. IOS Press. Amsterdam, The Netherlands. 1993; pp. 131-139.

[SoCR94] Soares, L.F.G.; Casanova, M.A.; Rodriguez, N.L.R. "Nested Composite Nodes and Version Control in Hypermedia Systems". *Proceedings of the Worshop on Versioning in Hypertext Systems*, in connection with ACM European Conference on Hypermedia Technology, Edinburgh. Setembro de 1994.

[SoCR95] Soares, L.F.G.; Casanova, M.A.; Rodriguez, N.L.R. "Nested Composite Nodes and Version Control in an Open Hypermedia System". *International Journal on Information Systems; Special issue on Multimedia Information Systems.* September 1995.

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

[SoSo95] Souza, G.L.; Soares, L.F.G. “Synchronization Aspects of a Presentation Model for Hypermedia Documents with Composite Nodes”. *Relatório Técnico PUC-RioInf.MCC31/95*. Rio de Janeiro, Brasil. Outubro 1995.