

Capítulo 4

Browsers e Trilhas para Documentos Hipermídia com Composições Aninhadas

A definição de documentos hipermídia é baseada geralmente em dois conceitos familiares chamados nós e elos. Existem alguns modelos conceituais que permitem agrupar nós e elos em nós de composição e também permitem composições aninhadas. Tais modelos hipermídia, permitindo que grupos de nós sejam organizados, hierarquicamente ou não, oferecem, entre outras facilidades, um grande suporte para a construção de ferramentas que ajudem a reduzir o problema da desorientação.

Um nó de composição agrupa nós e elos, chamados componentes, incluindo outros nós de composição, em modelos que permitem aninhamento. Os componentes podem estar ordenados, o que pode ser muito útil em alguns mecanismos de navegação, porém eles não formam necessariamente um conjunto, pois uma entidade pode ser incluída mais de uma vez no nó de composição. Alguns exemplos de sistemas hipermídia com nós de composição são HyperPro [Oste92], CoVer [Haak92] e HyperProp [SoCR95]. Em HyperPro, estes nós são chamados de nós de contexto e em CoVer e HyperProp de nós de composição. Outro importante exemplo é o modelo da proposta de padrão para codificação e representação de objetos multimídia/hipermídia, denominado MHEG [MHEG94].

Especializações da classe de nós de composição podem ser utilizadas em mecanismos de controle de versões. Muitos sistemas, com o objetivo de resolver o problema de manter a história de versões de um hiperdocumento, introduzem o conceito de grupos de versões. Exemplos destas estruturas podem ser vistos nos contextos de versões do HyperProp, nos “mobs” do CoVer e nos grupos de versões do HyperPro.

Os nós de composição também são úteis para modelar interações do usuário com os hiperdocumentos, tal como apresentado no Modelo Dexter [HaSc90]. Por este modelo, a camada de armazenamento contém informações estáveis, cujas instanciações constituem a camada de apresentação (*run time layer*). Uma instância, a apresentação de um componente da camada de armazenamento para o usuário, é o conceito fundamental desta camada. Instâncias existem dentro das sessões de trabalho do Modelo Dexter, como é mostrado na Figura 36.

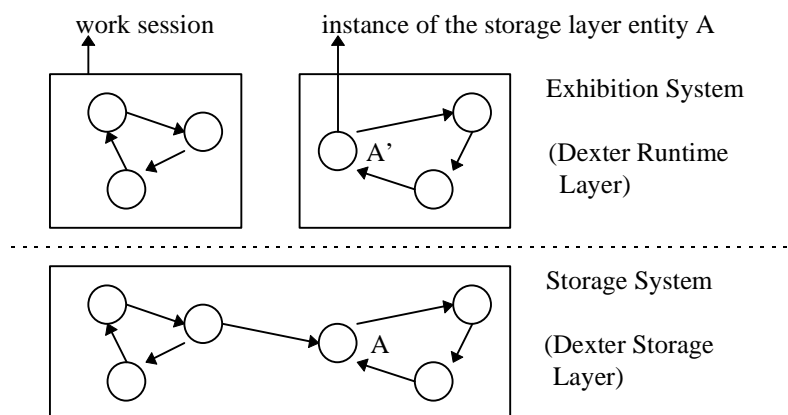


Figura 36: O Modelo Dexter para as camadas de armazenamento e apresentação

As sessões de trabalho e até o sistema de armazenamento inteiro podem ser modelados por nós de composição, como nos sistemas hipermídia CoVer e HyperProp. Sessões de trabalho são modeladas no HyperProp pelas bases privadas, e o sistema de armazenamento é modelado pela hiperbase pública, ambas especializações de nós de contexto. Uma instância no HyperProp é uma versão de um nó da hiperbase pública criada na base privada. Bases privadas, como uma especialização de nós de contexto, podem conter outras bases privadas, oferecendo suporte para organizar uma sessão de trabalho em várias sub-sessões aninhadas. A mesma idéia também pode ser encontrada nas “tasks” de CoVer.

Dentro do escopo de nós de composição, um browser é um mapa de seus componentes. Como uma ferramenta gráfica para navegação e edição em um hiperdocumento e dependendo do propósito do browser, pode-se ressaltar algumas características.

Se forem considerados nós de composição com o objetivo de ver somente o primeiro nível de aninhamento, deve-se considerar que a estrutura do nó é suficientemente simples para representá-la como uma visão ponto-a-ponto de todos os seus componentes, permitindo operações de navegação e edição. Este seria o mesmo caso das composições usadas para agrupar versões, também consideradas estruturas simples.

No caso do sistema de armazenamento ser visto em todos os níveis de detalhe de aninhamento, a composição correspondente terá certamente uma estrutura complexa de nós e elos, necessitando de métodos sofisticados para construir a ferramenta de visualização. Filtros para esconder informações e nós especiais, que sempre aparecem, provavelmente serão necessários. Felizmente a complexidade da ferramenta pode ser reduzida, porque usualmente não são permitidas operações de edição, somente de visualização.

No caso de um browser de uma sessão de trabalho de usuário ser vista em todos os níveis de detalhe de aninhamento, pode-se ter uma estrutura bastante complexa, na qual operações de edição e navegação devem ser permitidas. Logo, é necessária uma ferramenta com todas as características de ambos os browsers anteriormente mencionados.

O objetivo desta dissertação é construir uma ferramenta de navegação por browsers e trilhas para sistemas hipermídia baseados em modelos conceituais que suportam composições

aninhadas de nós [MuSC95]. Para validar a proposta de construção da ferramenta, foi feita uma implementação sobre o Modelo de Contextos Aninhados, que é o modelo conceitual sobre o qual o sistema HyperProp está baseado. O HyperProp auxiliará a construção de aplicações para o processamento de documentos, oferecendo um conjunto de ferramentas que permitirão portabilidade entre plataformas e bastante flexibilidade de uso. A proposta de construção de browsers é válida para todos os sistemas hipermídia que permitam composições aninhadas de nós, porém como a implementação foi feita sobre o sistema HyperProp, foi utilizado o Modelo de Contextos Aninhados e alguns de seus termos específicos para descrever o trabalho.

Neste capítulo, será apresentada uma breve descrição conceitual do Modelo de Contextos Aninhados e a proposta de construção de browsers e navegação por trilhas para o sistema HyperProp. Vale ressaltar mais uma vez que esta mesma proposta poderia ser utilizada por outros sistemas que permitam composições aninhadas de nós.

4.1 O Modelo de Contextos Aninhados

O Modelo de Contextos Aninhados (MCA) é um modelo conceitual hipermídia que, entre outros recursos, permite composições aninhadas de nós e elos, separa a estrutura dos documentos contida nos elos e âncoras das informações contidas nos nós, suporta conjuntos de versões, permite explorar e manusear configurações diferentes, mantém histórico dos documentos, suporta trabalho cooperativo e permite a definição de relações temporais e espaciais entre nós, para suas exibições.

A definição de documentos hipermídia no MCA é baseada em dois conceitos familiares, nós e elos. Nós são fragmentos de informação interligados por elos.

O modelo distingue duas classes básicas de nós, chamadas terminal e de composição. Um nó terminal pode conter dados em diversos tipos de mídia: texto, gráfico, áudio, vídeo etc. Um nó de composição agrupa nós e elos chamados componentes, incluindo outros nós de composição. Estes componentes não formam necessariamente um conjunto, pois uma entidade pode ser incluída mais de uma vez no mesmo nó. A classe de nós de composição pode ser especializada na classe de nós de contexto, que, por sua vez agrupa conjuntos de elos, nós terminais e nós de contexto e a classe trilha, que será discutida mais adiante. A classe de nós de contexto é especializada em cinco subclasses, que são: anotação, base privada, hiperbase pública, contexto de versões e contexto de usuário. Somente a classe de contexto do usuário pertence ao modelo básico do MCA, as outras são extensões para o controle de versões. Um nó de contexto do usuário agrupa um conjunto de elos, nós terminais e nós de contexto do usuário recursivamente. Esta classe, como qualquer nó de contexto, permite organizar um conjunto de nós hierarquicamente ou não. A Figura 37 resume a hierarquia de classes básicas do Modelo de Contextos Aninhados.

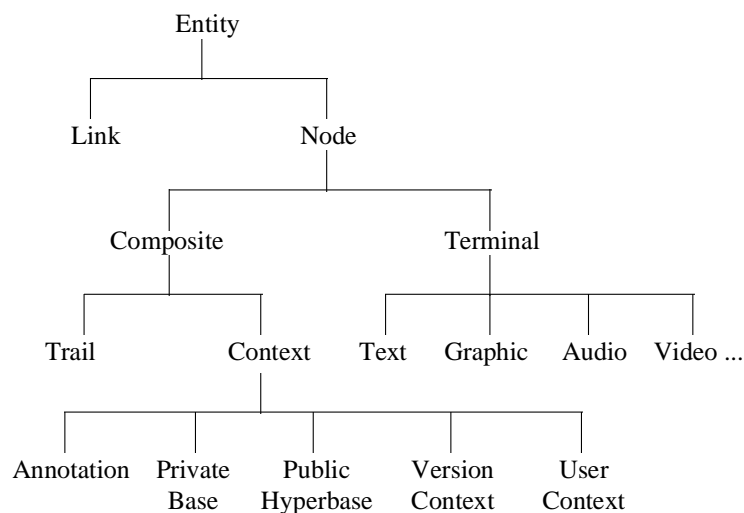


Figura 37: MCA - hierarquia de classes

Um elo conecta nós. Suas extremidades podem ligar nós inteiros ou regiões internas aos nós. A noção de região é descrita no modelo por âncora. Cada nó tem uma máscara que age como uma interface externa. Os elos referir-se-ão a regiões internas endereçando entradas na máscara do nó, chamadas âncoras. Cada elo contém um conjunto de pontos de origem e um conjunto de destinos. A multiplicidade de extremidades permite a definição de conexões “m-n”, dando suporte a aplicações onde a seleção de uma elo leva à exibição de vários nós. Os elos são sempre direcionais, apesar de poderem ser seguidos em ambas as direções. Elos podem ter condições e ações associadas. As ações são processadas nos nós destino se as condições forem satisfeitas.

O MCA também permite a definição de estruturas virtuais (nós, âncoras e elos), ou seja, que resultam da avaliação de alguma expressão. Uma estrutura virtual tem seu conteúdo computado quando é selecionada durante o processo de navegação.

O modelo de contextos aninhados permite que um mesmo nó esteja contido em diferentes nós de composição e que o aninhamento seja feito em qualquer nível. Daí, surge o conceito de perspectiva de um nó, que define por qual seqüência de nós de composição um determinado nó está sendo observado. Uma perspectiva para um nó N é uma seqüência $P=(N_m, \dots, N_1)$, $m \geq 1$, tal que $N_1 = N$ e N_{i+1} é um nó de composição que contenha N_i , para i que pertence a $[1, m)$. Podem existir diferentes perspectivas para um mesmo nó N , se ele estiver contido em mais de um nó de composição. A perspectiva corrente de um nó é a dada pela última navegação ao nó.

Um trilha é uma especialização de nó de composição que contém uma lista ordenada de nós, incluindo trilhas. Um nó pode estar na lista em mais de uma posição.

Para suportar controle de versões, a classe de nós de contexto pode ser especializada em quatro outras classes: anotação, hiperbase pública, base privada, contexto de versão e contexto de usuário.

O contexto de versão agrupa versões de um mesmo objeto, permitindo manter um histórico do documento. A relação entre as versões é dada através de elos, que, por sua vez, não podem ter versão. Apenas nós terminais e de contexto são versionáveis.

Para suportar trabalho cooperativo e permitir a criação automática de versões, é necessário garantir a consistência entre nós relacionados, o que é feito definindo-se estados para os nós. Estes podem estar nos estados temporário, permanente ou obsoleto.

Um nó fica no estado temporário quando é criado e assim permanece enquanto estiver sendo modificado. Quando se torna estável, é promovido ao estado permanente explicitamente por pedido do usuário ou implicitamente por operações que o modelo oferece. Um nó no estado permanente não pode ser alterado ou eliminado diretamente pelo usuário, mas apenas promovido a obsoleto, quando deve ser notificado aos outros nós que fazem referência a ele.

O conceito de estado de um nó só é relevante para contextos do usuário e nós terminais que possuem atributos versionáveis. Quando diz-se que nós permanentes não podem ser alterados, significa que seus atributos versionáveis não podem ser modificados.

No suporte a trabalho cooperativo, deve-se permitir o compartilhamento de informações, e ao mesmo tempo, prover mecanismos para criar informações privadas. No MCA, isto é fornecido através de dois conceitos: hiperbase pública e base privada.

Uma hiperbase é qualquer conjunto de nós, tal que, se contém um nó de contexto, então todos os seus nós componentes também pertencem à hiperbase.

A hiperbase pública corresponde a informações públicas, estáveis. Por isso, só pode conter nós permanentes ou obsoletos. Enquanto que uma base privada contém as entidades utilizadas durante uma sessão de trabalho do usuário. Uma base privada pode conter outras bases privadas, permitindo a organização das sessões de trabalho em subssões aninhadas.

Definindo mas precisamente, a hiperbase pública é um tipo especial de nó de contexto que agrupa conjuntos de nós terminais e nós de contexto de usuário. Se um nó de composição C está na hiperbase, então, todos os nós componentes de C também estão.

Uma base privada é um tipo especial de nó de contexto que agrupa qualquer entidade, exceto a hiperbase pública e nós de contexto de versões, tal que:

- uma base privada deve estar contida em pelo menos uma base privada;
- se um nó de composição N está contido em uma base privada BP, seus componentes estão contidos em BP ou na hiperbase pública ou em qualquer base privada recursivamente contida em BP.

Um nó não pode ser movido da hiperbase pública para uma base privada, o usuário deve criar uma versão deste nó. Porém o processo inverso é permitido, obrigando que o nó passe para o estado permanente, se ainda não estiver neste estado.

Facilidades para que o usuário comente determinado trabalho são providas pelas anotações. Estas podem ser feitas em qualquer formato de mídia e guardam referências às versões anotadas. As anotações só podem ser incluídas em bases privadas.

Para maiores detalhes sobre o modelo, deve-se consultar [CTLR+91], [CCLS92], [SoCa93], [SoCC93], [SoCR93a], [SoCR93b], [SoRC93], [SoCR95].

4.2 Browsers e Trilhas para o MCA

A motivação principal desta dissertação é construir uma ferramenta gráfica que auxilie o usuário na navegação pelos diversos nós do(s) documento(s), agrupados em contextos definidos pelo MCA, tentando minimizar ao máximo o problema da desorientação.

Para isso, a idéia é fornecer ao usuário informações de âmbito espacial e temporal, para que ele possa identificar onde ele está, como chegou lá e para onde pode ir.

O objetivo é oferecer estas informações através de dois mecanismos de navegação de hipertextos: browsers e trilhas. As informações de localização na estrutura de contextos aninhados e de quais direções podem ser seguidas são dadas pelos browsers e o histórico do caminho percorrido é obtido pelas trilhas. Além de trilhas que registram o caminho do usuário, existirão outras que orientarão os leitores, sugerindo maneiras de navegar pelos nós do documento, ou sua linearização.

Como o propósito do HyperProp é construir ferramentas com bastante flexibilidade, o browser permitirá que o usuário escolha o nível de detalhe a ser apresentado no diagrama, influenciando no número de nós e elos exibidos. Isto permitirá que o usuário molde a complexidade do diagrama de acordo com seus interesses. Assim, em determinado momento, toda a rede de interconexões poderá ser visualizada, enquanto que em outro instante, apenas o mínimo necessário para fornecer ao usuário algum senso de localização.

Como browsers e trilhas são ferramentas de auxílio à navegação no hipertexto, estes recursos permitirão que o usuário navegue pelos nós e elos através de representação gráfica, visualizando o conteúdo dos nós e, em determinados casos, até editando o documento. As próximas seções apresentam uma descrição do mecanismo de browsers e trilhas do HyperProp [MuSo95].

4.2.1 Navegação por Browsers

O sistema HyperProp oferece três tipos de browsers: o browser de contexto, o browser de hiperbase e o browser de base. Esta divisão é necessária porque cada componente de um modelo hipermídia com nós de composição aninhados possui determinadas características e oferece determinados recursos de navegação e edição ao usuário.

No browser de contexto, será possível visualizar a estrutura de um nó de contexto através da visão ponto a ponto dos seus nós componentes e elos, segundo a perspectiva corrente. Isto é, serão exibidos todos os nós e elos pertencentes ao contexto e os elos visíveis a partir da perspectiva corrente. Neste browser serão permitidas operações de navegação e edição gráfica do contexto, através da inclusão, eliminação e alteração de nós e elos. A edição de elos só será realizada para os elos que pertencerem ao contexto sendo visto, sem considerar os que foram herdados pela perspectiva corrente. Este browser estará disponível para nós de contexto do usuário e de contexto de versões.

O browser de hiperbase apresentará uma visão um pouco diferente da rede de interconexões, pois a hiperbase deverá ter uma estrutura bem mais complexa que um nó de contexto. Devido a este fato, será necessária a utilização de filtros na exibição da rede de nós e elos. Estes filtros têm o objetivo de melhorar a legibilidade dos diagramas, mostrando somente o que é de interesse do usuário. Para isto, o próprio usuário deverá ser capaz de escolher o nível de detalhe apresentado de acordo com sua necessidade. Este browser não permitirá operações de edição do documento, somente a visualização da estrutura, e estará disponível para a hiperbase pública.

O browser de base agrupará características do browser de contexto e do browser de hiperbase. Ele permitirá navegação e edição gráfica do contexto, exibindo seu conteúdo através da utilização de filtros que visam melhorar a legibilidade do diagrama de nós e elos. Este browser estará disponível para as bases privadas.

A opção de se adotar três tipos diferentes de browsers deve-se ao fato de que o MCA, através do aninhamento de contextos, permite que se crie documentos bem estruturados. Baseado nisto, em determinado nó de contexto de usuário ou de versão não se deve ter uma estrutura muito complexa, tornando viável visualizar todos os seus componentes sem a utilização de filtros que omitam informação. Isto diminui o overhead para a apresentação do diagrama e permite a edição dos seus elementos, já que todo o conteúdo do contexto está sendo visto. Já a rede de interconexões da hiperbase pública e das bases privadas pode ser bastante complexa, o que exige mecanismos para melhorar a legibilidade da representação gráfica. Por outro lado, a edição do documento é restrita aos nós de contexto do usuário e de versões e às bases privadas, não podendo ser feita na hiperbase pública. Por estas razões, é interessante manter três tipos de browsers, um que permita edição e tenha um mecanismo de exibição mais simples, outro em que a edição não é permitida, mas que forneça maneiras de controlar a complexidade do diagrama e outro que combine todas estas características.

A navegação através do browser pode ser feita por nós e por elos, o que oferece bastante flexibilidade ao usuário.

Com o propósito de orientar o usuário sobre que nós já foram navegados, todos os nós visitados são exibidos de forma diferenciada no diagrama, permitindo ao usuário adquirir uma noção de que parte do documento já foi acessado.

Caso o diagrama exibido tenha muitos elos, dificultando a sua legibilidade, o usuário poderá optar por esconder todos os elos, só visualizando a hierarquia de contextos.

Como o próprio MCA permite que um nó de contexto agrupe vários outros nós de contexto ou nós terminais no diagrama exibido, um determinado nó, mostrado por um ícone, pode representar diversos nós da estrutura do documento. Esta idéia vai de encontro ao que é oferecido pelo CYBERMAP (Seção 3.4), onde um ícone representa vários nós. A diferença é que o CYBERMAP exige um processamento prévio para agrupar os nós mais relacionados e o MCA já fornece este agrupamento pela própria estrutura de contextos.

A hiperbase pública e as bases privadas serão tratadas como grandes nós de contexto nos browsers, ou seja, para o algoritmo de construção do diagrama, elas serão o contexto mais externo cujo browser está sendo visualizado. No caso das bases privadas, será permitida a edição do conteúdo da base ou de seus componentes através do browser de base.

4.2.1.1 Browser de Contexto

Como já mencionado, o browser de contexto será utilizado para os nós de contexto do usuário e de contexto de versões. Seu propósito é fornecer ao usuário uma ferramenta para edição através da forma gráfica de cada contexto específico, tendo um mecanismo de exibição da rede de nós e elos mais simples do que o do browser de hiperbase ou de base.

O diagrama exibirá o primeiro nível de aninhamento dos contextos, sem se preocupar com informações internas aos nós componentes do contexto que está sendo observado. Este browser só exibirá os elos visíveis a partir do contexto sendo visualizado, mostrando os que pertencem explicitamente ao contexto e os que foram herdados a partir da perspectiva, desenhando-os de forma diferenciada.

Neste browser, não será utilizado nenhum tipo de filtro para omitir informações, pois para a edição do contexto é necessário ter noção de toda a sua estrutura.

Serão permitidas as operações de criação, inserção e eliminação de nós e elos do documento de forma gráfica, o que facilita a definição da sua rede de interconexões. Porém, a edição deve ser feita no browser do contexto específico desejado, logo, a edição de elos só será permitida para os que pertencerem explicitamente ao contexto, não sendo possível editar elos herdados a partir da perspectiva corrente.

Caso o usuário queira criar um elo que tenha como extremidade um nó na estrutura interna de um dos componentes do contexto em foco, ele terá que “explodir” os contextos sucessivamente até atingir o nó origem ou destino do arco. Por exemplo, seja A um nó de contexto que contém B, que por sua vez contém C e D. Para definir um elo em B que ligue C e D tem-se que ativar o browser de B e criar o elo $(\langle C, i \rangle, \langle D, j \rangle)$, onde i e j são âncoras de C e D respectivamente (vide Figura 38-a). Porém, se o desejado é criar um elo no contexto A conectando C e D, deve-se ativar a operação de criação de elo em A, e navegar em profundidade até o contexto B para definir as extremidades do elo, que passa a ser $(\langle B, (C, i) \rangle, \langle B, (D, j) \rangle)$ [SoRC93] (vide Figura 38-b). Ainda existe uma terceira forma de criar o elo que liga C e D através da criação de três elos: o primeiro ligando C a B $(\langle C, i \rangle, \langle B, j \rangle)$; o segundo ligando B a B $(\langle B, j \rangle, \langle B, k \rangle)$; e o terceiro ligando B a D $(\langle B, k \rangle, \langle D, l \rangle)$ (vide Figura 38-c). No primeiro caso, o contexto em foco deve ser B, não

necessitando visualizar outro contexto para definir o elo. Já no segundo caso, tem-se que “explodir” o contexto B componente de A para definir o elo que liga C e D. A “explosão” significa navegar em profundidade até o contexto B, vindo do contexto A. E no terceiro caso, tem-se que visualizar o contexto A para criar os elos ligando *B a B* e os elos que ligam *C a B* e *B a D*, compondo finalmente o elo C a D.

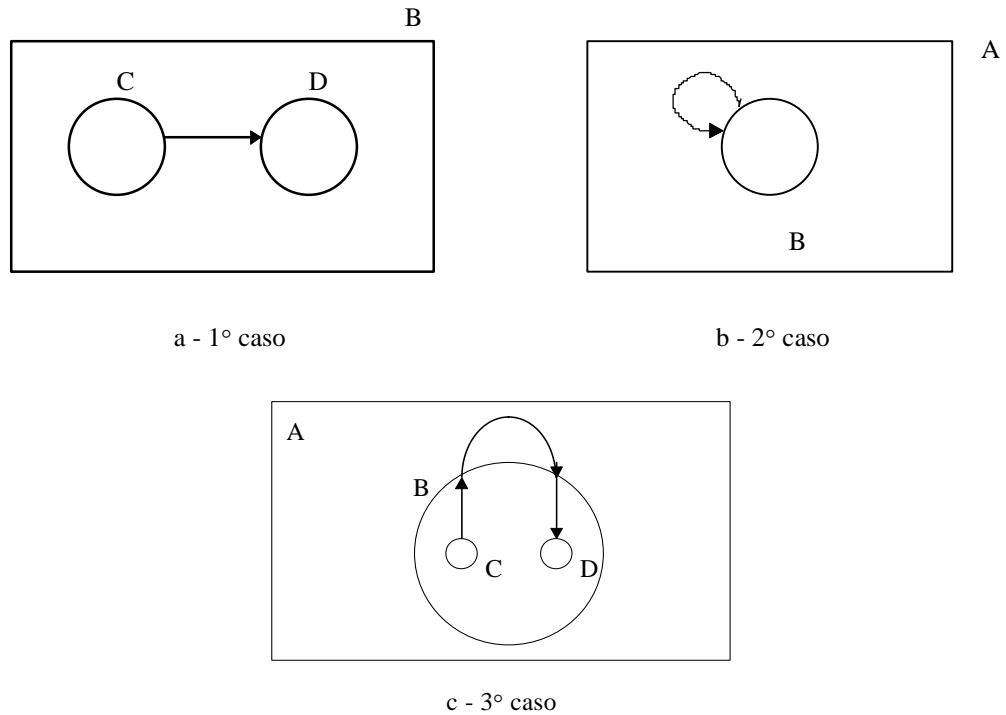


Figura 38: Criando elos em A e em B

Se o elo que liga C e D for definido no contexto A (segundo caso do exemplo), ele será exibido quando o contexto B for visualizado, mas sua edição não será permitida, pois não pertence explicitamente a B (elo 2 da Figura 39).

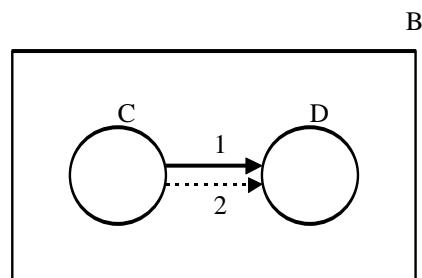


Figura 39: Browser de B: elo 1, definido em B e elo 2, definido em A.

4.2.1.2 Browser de Hiperbase

O browser de hiperbase será utilizado para visualizar a hiperbase pública, por isso, concentrará sua atenção na exibição da estrutura da rede de nós e elos. Seu objetivo é facilitar o entendimento do leitor em relação à sua posição nos contextos local e global dos documentos, tentando diminuir o problema da desorientação.

Para o browser de hiperbase, a complexidade da estrutura de contextos influirá no conteúdo do diagrama a ser exibido, ou seja, quanto mais complexa a rede de nós e elos, mais informação será filtrada. Isto porque não adianta mostrar para o usuário uma visão ponto a ponto de uma estrutura complexa, pois isso só prejudica o entendimento da organização do documento, como já foi experimentado por outros projetos de sistemas hipermídia, como o Intermedia [UtYa89], por exemplo. Logo, para a exibição de redes maiores, é necessário o uso de filtros que escondam informações que o usuário não teria interesse em visualizar em determinado momento. Mesmo assim, é válido oferecer ao usuário a possibilidade de moldar o filtro de acordo com seus interesses, ou melhor, de estabelecer se mais ou menos informação deve ser omitida.

Para a construção desses filtros, utiliza-se uma extensão do modelo de lentes olho-de-peixe [Furn86], para nós definidos pelo modelo de contextos aninhados, e alguns conceitos do CYBERMAP adaptados. Com isso, o usuário terá tanto informações de interesse local, que estão bastante ligadas ao nó em foco, como informações de âmbito global, que dão idéia da posição atual em relação à estrutura de contextos do documento.

A estratégia do olho-de-peixe, como visto, propõe a definição de uma função do grau de interesse do usuário de cada nó em relação ao nó em foco. Esta função é dada pela importância a priori do nó e a sua distância em relação ao nó focado.

Para o Modelo de Contextos Aninhados, cada nó exibido representa, na verdade, a perspectiva em questão, pois o grau de interesse do usuário em relação a cada nó depende da perspectiva em que está sendo analisado. Então, um mesmo nó terá um grau de interesse para cada uma de suas perspectivas. Isso faz com que o nó possa ser exibido mais de uma vez ou possa ser visualizado por uma perspectiva em determinado momento e não esteja necessariamente visível em outra, pois a visibilidade depende de cada perspectiva do nó.

Com isso, a função de grau de interesse é calculada para cada nó, considerando-se cada uma de suas perspectivas. Na verdade, para o cálculo da função, cada ocorrência do nó na estrutura de composições é considerada como uma entidade distinta, ou seja, cada ocorrência do nó terá um valor de grau de interesse do usuário.

Assim sendo, as componentes da função de grau de interesse podem ser definidas da seguinte maneira:

- a importância a priori do nó x , em cada uma de suas perspectivas, denotada por $API(x)$, é dada pelo valor negativo do seu nível de aninhamento em relação ao contexto mais externo (hiperbase). Quanto maior o nível de aninhamento, menor a importância a priori do nó, naquela perspectiva;

- a distância do nó x para o nó em foco y , considerando-se a perspectiva corrente de y , denotada por $D(x,y)$ é calculada, para cada uma das perspectivas de x , através da seguinte função:

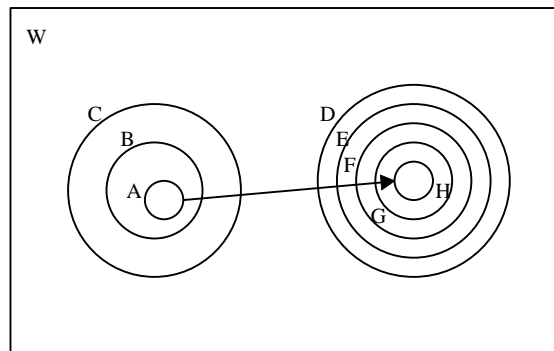
$$D(x,y) = \min (Dc(x,y), De(x,y))$$

onde $Dc(x,y)$ é a distância entre x e y considerando somente navegação por contextos e $De(x,y)$ é a distância mínima entre x e y considerando somente navegação por elos. Se não existirem elos que liguem x a y , a distância $D(x,y)$ será igual a distância em contextos $Dc(x,y)$.

Para se calcular a distância em contextos $Dc(x,y)$, navega-se em profundidade do nó x até atingir o nó y , contando quantos contextos foram percorridos.

Para se calcular a distância em elos $De(x,y)$, navega-se por elos com extremidade em y até atingir x , contando todos os contextos por que os elos passam, inclusive os que fazem parte do caminho do elo. Faz-se uma navegação exaustiva até esgotarem-se todos os caminhos entre x e y navegando-se por elos. A distância $De(x,y)$ é o caminho mínimo entre x e y , considerando-se navegação por elos. Nota-se que os elos são considerados bidirecionais.

Para ilustrar o cálculo da distância entre dois nós, considere o seguinte exemplo:



$$D(A,H) = \min (Dc(A,H), De(A,H))$$

$$Dc(A,H) = 8 \text{ (caminho: B-C-W-D-E-F-G-H)}$$

$$De(A,H) = 7 \text{ (caminho: B-C-D-E-F-G-H)}$$

$$\text{logo, tem-se } D(A,H) = \min (8,7) = 7$$

Calculando o grau de interesse de cada nó, para cada uma de suas perspectivas, de acordo com a função:

$$DOI(x,y) = API(x) - D(x,y)$$

serão exibidas as ocorrências dos nós na estrutura de composições que tiverem grau de interesse maior ou igual a determinado valor K , especificado pelo usuário.

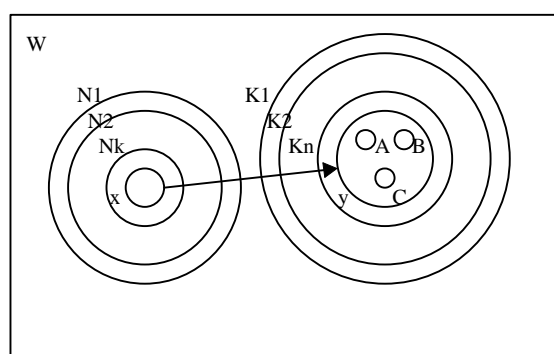
Com isso, são fornecidos meios de visualização da estrutura em vários níveis de detalhe, não necessitando de um novo processamento para recalcular o grau correspondente a cada nó quando o nível de interesse for modificado. Este só deverá ser recalculado quando o nó em foco mudar, afetando a distância entre os nós, ou quando houver alterações na estrutura de aninhamento dos contextos, afetando a importância a priori dos nós alterados.

É importante que alguns requisitos sejam satisfeitos pelo browser em qualquer nível de detalhe. Entre estes requisitos, pode-se destacar:

- se um nó é exibido no diagrama, sua perspectiva também é exibida, garantindo que a noção de contextos não seja perdida;
- todos os nós x diretamente ligados ao nó em foco y aparecem sempre no diagrama;
- para elos do tipo $(\langle N_1, N_2, \dots, N_k, x \rangle, i, \langle K_1, K_2, \dots, K_n, y \rangle, j)$, onde o nó y é o foco, o nó N_1 é sempre exibido e os nós N_1, N_2, \dots, N_k têm prioridade decrescente de exibição;
- se um nó de contexto é o nó em foco y , seus componentes aparecem sempre no diagrama.

Deve-se ressaltar que o algoritmo fornecido para calcular a função de grau de interesse satisfaz todos os requisitos acima, como será mostrado a seguir.

Considere uma estrutura de contextos genérica, com foco em y :



A, B e C são nós quaisquer componentes de y .

As seguintes funções definem o grau de interesse dos nós:

$API(W) = 0$		
$API(N1) = -1$	$API(K1) = -1$	$API(A) = -n-2$
$API(N2) = -2$	$API(K2) = -2$	$API(B) = -n-2$
...	...	$API(C) = -n-2$
$API(Nk) = -k$	$API(Kn) = -n$	
$API(x) = -k-1$	$API(y) = -n-1$	
$D(W,y) = n+1$		
$D(N1,y) = n+1$	$D(K1,y) = n$	$D(A,y) = 1$
$D(N2,y) = n+2$	$D(K2,y) = n-1$	$D(B,y) = 1$
...	...	$D(C,y) = 1$

$$\begin{array}{ll} D(Nk,y) = n+k & D(Kn,y) = 1 \\ D(x,y) = n+k+1 & D(y,y) = 0 \end{array}$$

e como $DOI(x,y) = API(x) - D(x,y)$

$$\begin{array}{lll} DOI(W,y) = -n-1 & & \\ DOI(N1,y) = -n-2 & DOI(K1,y) = -n-1 & DOI(A,y) = -n-3 \\ DOI(N2,y) = -n-3 & DOI(K2,y) = -n-1 & DOI(B,y) = -n-3 \\ ... & ... & DOI(C,y) = -n-3 \\ DOI(Nk,y) = -n-2k & DOI(Kn,y) = -n-1 & \\ DOI(x,y) = -n-2(k+1) & DOI(y,y) = -n-1 & \end{array}$$

a) Garante-se que a perspectiva de um nó qualquer x sempre aparece se x for exibido, pois

$$DOI(x,y) \leq DOI(Nk,y) \leq \dots \leq DOI(N2,y) \leq DOI(N1,y)$$

Logo, para $DOI(x,y) \geq k$, $DOI(Ni,y) \geq k$, para todo $i > 0$.

b) Garante-se que todos os nós diretamente ligados ao nó em foco y são sempre exibidos se o maior valor de k for (-n-2). Na verdade o maior valor possível de k = -n-1, porém, só seria exibida a perspectiva corrente de y. Como é importante que os nós diretamente conectados a y sempre sejam exibidos, faz-se o maior valor de k = -n-2, não permitindo que k exceda este valor.

c) Garante-se que o nó N1 é exibido e os nós N2,N3,...,Nk têm prioridade decrescente de exibição, pois

$$DOI(N1,y) > DOI(N2,y) > \dots > DOI(Nk,y)$$

e $DOI(N1,y) = -n-2$ que é o maior valor possível de K, como dito no item (b).

d) Garante-se que os componentes A,B,C do nó em foco y sempre aparecem no diagrama fazendo o maior valor de K = -n-3, na verdade, o maior valor de K seria -n-1, porém, para satisfazer os requisitos (b) e (d), K não deve exceder -n-3.

Então, com o objetivo de garantir a satisfação destes requisitos pelo browser, deve-se limitar o valor máximo de K em:

$$\begin{array}{l} API(foco) - 2, \text{ para foco em nós de composição;} \\ API(foco) - 1, \text{ para foco em nós terminais.} \end{array}$$

Como o grau de interesse depende do nó em foco, quando este for modificado, passando o foco para outro nó, pode ser que alguns contextos se fechem e outros sejam abertos no diagrama. Isto vai depender do valor da função $DOI(x,y)$ do contexto x em relação ao nó em foco y do browser. Este fenômeno de abrir e fechar contextos é que garantirá a legibilidade do diagrama exibido, pois a idéia de representar um conjunto de nós por apenas um ícone no diagrama é fornecida implicitamente pela definição de contextos do MCA. Esta

idéia é semelhante a oferecida pelo CYBERMAP, sendo que no MCA, não é necessário esforço computacional para encontrar semelhança entre os nós, isto já está implícito no agrupamento de contextos.

A importância a priori de um nó será calculada quando o nó é inserido em determinado contexto, existindo uma importância para cada perspectiva do nó.

A distância de cada nó x ao nó em foco y será calculada quando o browser da hiperbase é ativado ou quando o foco y é modificado.

Para desenhar o diagrama da hiperbase utiliza-se a notação de conjuntos aninhados [Knut73]. Esta forma de representação gráfica expressa a hierarquia de contextos através de conteúdo espacial [Fein88], distinguindo a noção de aninhamento das ligações entre os nós, o que fornece uma visão adequada da estrutura do hipertexto.

Nem todos os nós contidos recursivamente na composição serão considerados para o cálculo da função DOI, mas somente os seus componentes diretos e os nós que já foram acessados na sessão de trabalho em questão. Isto melhora consideravelmente a legibilidade do diagrama, pois o usuário só visualizará mais detalhes sobre os nós que realmente lhe interessam, logo o diagrama fica bem mais limpo. O caso em que todos os nós são considerados para o cálculo da função de grau de interesse é um caso específico, onde o usuário já acessou todos os contextos na sessão de trabalho.

Da mesma forma que um nó é considerado para o cálculo da função DOI no momento em que um usuário o acessa, este mesmo nó pode ser desconsiderado através de um pedido do usuário. Isto permite que o usuário elimine o nó do diagrama a ser construído.

Como a hiperbase pública contém todos os nós inclusive os nós componentes dos nós de contexto da hiperbase, em um primeiro momento, o browser da hiperbase mostra todos os seus nós, sem exibir detalhes internos a estes nós. À medida que o usuário escolhe o nó de interesse, o escopo muda da hiperbase para o nó escolhido, fazendo com que este nó seja o contexto mais externo e os outros nós da hiperbase sejam ignorados para a exibição do browser. Pode-se especificar uma marcação especial para nós, como um componente de sua API, que implique que estes nós serão visualizados quando o browser de hiperbase é ativado. Isto reduziria o número de nós exibidos na visualização inicial do browser da hiperbase pelo usuário. Uma vez escolhido o documento e navegando em profundidade, este componente da API deve ser ignorado.

A posição de cada nó no browser de hiperbase depende obviamente de quais nós são exibidos em determinado momento, logo os nós não têm uma posição fixa na tela. Por isso será dada ao usuário a opção de definir certos nós como pontos de referência, ou seja, estes nós específicos serão sempre exibidos no diagrama, ajudando o usuário a se localizar no contexto espacial. Estes nós, chamados pontos de referência (landmarks), serão discutidos com mais detalhes na subseção seguinte.

4.2.1.2.1 Landmarks

Landmarks são perspectivas que servem como pontos de referência para o usuário. Por terem importância significativa na rede de nós e elos, estarão sempre presentes no diagrama, auxiliando a orientação do usuário.

Estes pontos de referência serão definidos de acordo com o perfil do usuário, pois a escolha de quais perspectivas são landmarks pode não ser a mesma para todos os usuários.

Como a estrutura de nós do MCA é definida através de contextos, os nós componentes das perspectivas landmarks serão exibidos dependendo do nível de aninhamento sendo observado. Isto é, um nó N que é integrante de uma perspectiva landmark e tem importância a priori -2, ou seja, faz parte do segundo nível de aninhamento, será exibido no diagrama se qualquer outro nó de importância -2 for exibido, mesmo se o grau de interesse de N em relação ao nó em foco não for suficiente para marcá-lo para exibição. Entretanto, uma perspectiva landmark é sempre exibida no diagrama, sendo representada por, pelo menos, um de seus nós integrantes (aquele com importância a priori igual a -1).

Com isso, definimos o conjunto de landmarks de uma dada composição pelo conjunto de perspectivas landmarks definidas explicitamente pelo usuário unido ao conjunto de prefixos destas perspectivas.

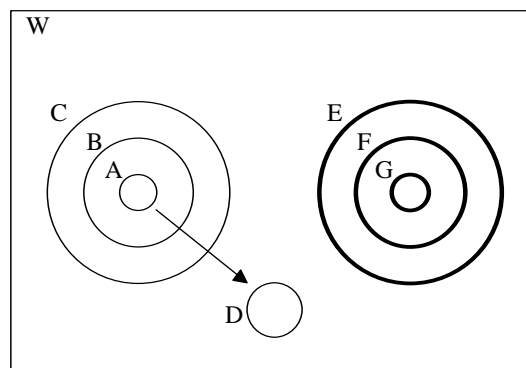
Se uma perspectiva x for landmark, a função de cálculo do grau de interesse para cada um de seus nós componentes será definida por:

$$DOI(x,y) = LM(x)$$

onde $LM(x)$ é um valor que garante que x seja exibido, satisfazendo à condição $DOI(x,y) \geq K$. Sendo assim, a função $DOI(x,y)$, quando x é componente de uma perspectiva landmark, é recalculada cada vez que o valor de K é modificado, pois $LM(x)$ é modificado.

Os landmarks serão exibidos de forma diferenciada no diagrama, fazendo com que possam ser identificados facilmente pelo usuário.

Exemplo: Considere a composição W, tendo o nó D como foco e como landmark a perspectiva (W, E, F, G)



Definindo a função $API(x)$:

$API(A) = -3$	$API(E) = -1$
$API(B) = -2$	$API(F) = -2$
$API(C) = -1$	$API(G) = -3$
$API(D) = -1$	$API(W) = 0$

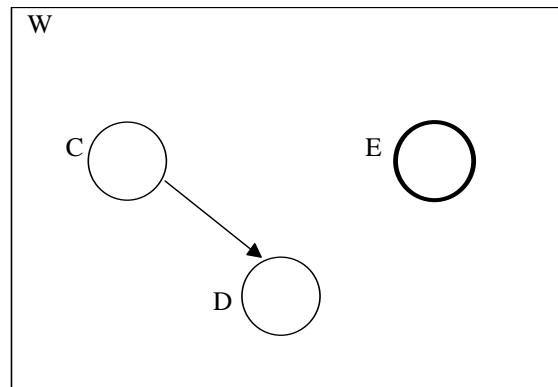
Definindo a distância $D(x,D)$:

$D(A,D) = 3$
$D(B,D) = 2$
$D(C,D) = 1$
$D(D,D) = 0$

Calculando a função de grau de interesse $DOI(x,D)$:

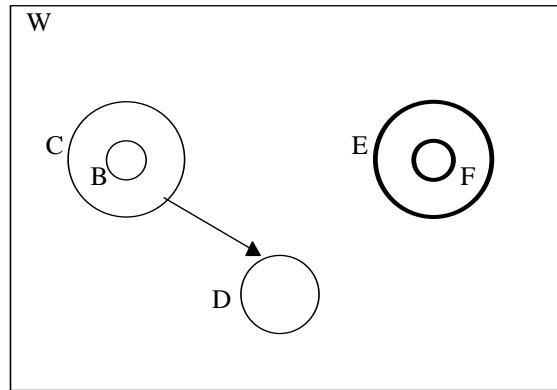
$DOI(A,D) = -6$	$DOI(E,D) = LM(E)$
$DOI(B,D) = -4$	$DOI(F,D) = LM(F)$
$DOI(C,D) = -2$	$DOI(G,D) = LM(G)$
$DOI(D,D) = -1$	$DOI(W,D) = LM(W)$

Desenhando o diagrama para $DOI(x,D) \geq -2$, temos:



O nó E aparece no diagrama porque é integrante de uma perspectiva landmark. Como $DOI(x,D) \geq -2$, $LM(E) = -2$. Os nós F e G não são exibidos porque só estão sendo exibidos nós com $API \geq -1$.

Desenhando o diagrama para $DOI(x,D) \geq -4$, temos:



Os nós E e F aparecem no diagrama porque são integrantes de uma perspectiva landmark. Como $DOI(x,D) \geq -4$, $LM(E) = -4$ e $LM(F) = -4$. O nó G não é exibido porque só estão sendo exibidos nós com $API \geq -2$.

Para $DOI(x,D) \geq -6$, tem-se o diagrama original e o valor da função $LM(x)$ é -6 para os nós E, F e G.

4.2.1.3 Browser de Base

O browser de base será utilizado para as bases privadas e permitirá tanto a edição gráfica como a navegação. Como a rede de interconexões das bases privadas tende a ser razoavelmente complexa, o mecanismo de visão com olho-de-peixe também será utilizado para filtrar algumas informações.

Este browser utilizará o mesmo algoritmo de cálculo do grau de interesse apresentado no browser de hiperbase.

Deve-se ressaltar que a todas as operações de edição de bases privadas descritas no Modelo de Contextos Aninhados estão disponíveis e também algumas operações para os seus nós de contextos componentes.

No browser de base, 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 em nós componentes da base.

Além destas operações sobre o conteúdo da base privada e seus componentes, existem outras operações que foram definidas no modelo conceitual para realizar a transferência de nós entre as bases privadas e a hiperbase pública. São elas:

- **Check-out:** utilizada para mover um nó da base privada para a hiperbase pública. Se um nó de contexto de usuário C for movido da base privada para a hiperbase pública, então, todos os nós terminais e de contexto de usuário componentes de C também são movidos para a hiperbase;

- **Shift:** utilizada para mover todos os nós terminais e de contexto de usuário da base privada para a hiperbase pública. Se uma base privada possuir outras bases privadas, o conteúdo destas últimas também será transferido para a hiperbase pública. No fim do processo de *shift*, a base privada só conterá trilhas, anotações (e elos associados) e bases privadas, que só conterão trilhas, anotações e bases privadas recursivamente;
- **Open:** utilizada para criar versões de nós terminais e de contexto de usuário da hiperbase pública em uma base privada. A primitiva *open*, cria versões de todos os nós componentes recursivamente do nó original, se este for um nó de contexto de usuário;
- **Check-in:** utilizada para criar versões de nós terminais e de contexto de usuário da hiperbase pública em uma base privada. Esta primitiva só cria a versão do nó escolhido, sem criar versões de seus componentes a princípio, o que é feito à medida que o usuário acessa os nós componentes.

Para visualizar melhor o algoritmo de construção do diagrama, será apresentado um exemplo de estrutura de contextos, indicando como o diagrama exibido pode variar dependendo do nó em foco ou do grau de interesse escolhido pelo usuário, utilizando o algoritmo do olho-de-peixe.

4.2.1.4 Exemplo de Browser com Olho-de-Peixe

Seja A uma composição com a seguinte estrutura de nós e elos:

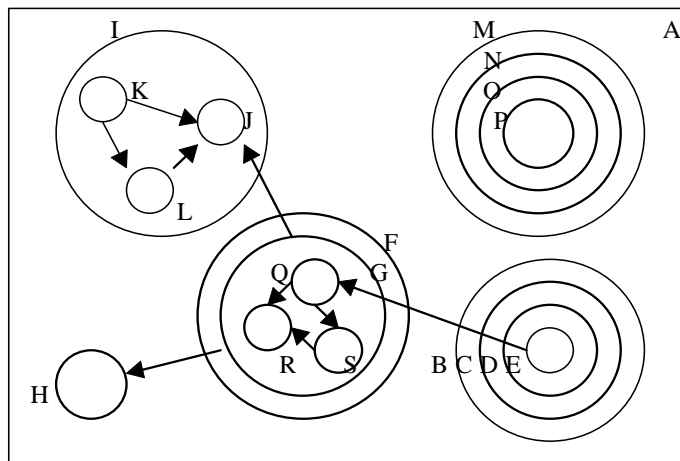
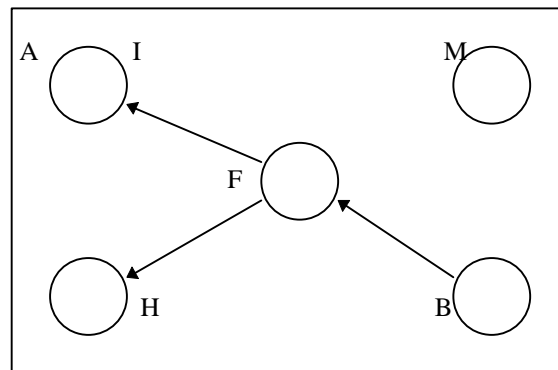


Figura 40: Exemplo de composição A

Inicialmente só serão exibidos os nós componentes diretos de A, ou seja, (B,F,H,I,M). À medida que o usuário acessa cada um desses nós, o nó em foco é definido e a lista de nós para cálculo da função DOI é aumentada. Por exemplo, imagine que o nó F seja acessado, a nova lista para computar a função DOI é (A,B,F,G,H,I,M). Novamente, se o nó G for acessado, a lista será (A,B,F,G,H,I,M,Q,R,S). Se o nó I for acessado, a nova lista será (A,B,F,G,H,I,J,K,L,M,Q,R,S), e assim por diante.

Deve-se notar que apesar do nível de detalhe ser o mesmo (o valor de K permanece inalterado), o conteúdo do diagrama pode ser modificado, de acordo com a navegação do usuário pelos nós do documento.

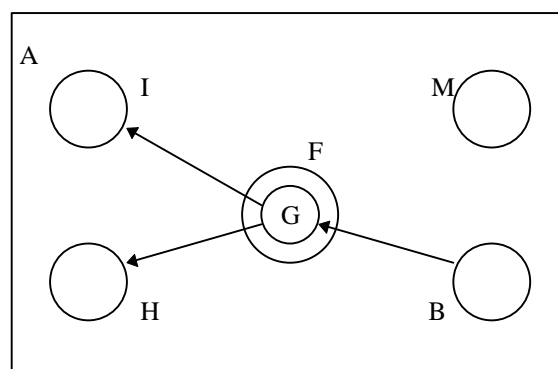
Quando o browser da composição A é ativado, tem-se o seguinte diagrama:



Se o nó F for acessado, calcula-se a função DOI (foco em F):

Nó	API	D	DOI
A	0	1	-1
B	-1	2	-3
F	-1	0	-1
G	-2	1	-3
H	-1	2	-3
I	-1	2	-3
M	-1	2	-3

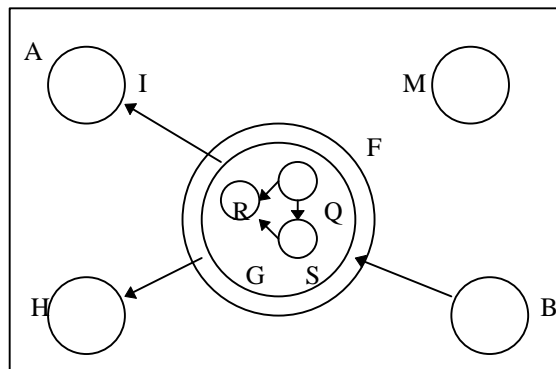
O diagrama será então:



Se o nó G for acessado, tem-se para a função DOI (foco em G):

Nó	API	D	DOI
A	0	2	-2
B	-1	3	-4
F	-1	1	-2
G	-2	0	-2
H	-1	2	-3
I	-1	2	-3
M	-1	3	-4
Q	-3	1	-4
R	-3	1	-4
S	-3	1	-4

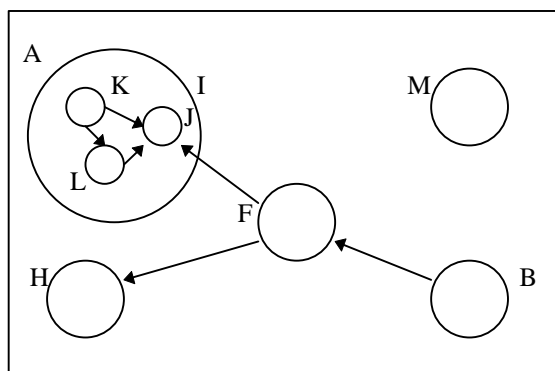
Com isso, tem-se o seguinte diagrama:



Se o foco mudar para o nó I, teremos:

Nó	API	D	DOI
A	0	1	-1
B	-1	2	-3
F	-1	2	-3
G	-2	3	-5
H	-1	2	-3
I	-1	0	-1
J	-2	1	-3
K	-2	1	-3
L	-2	1	-3
M	-1	2	-3
Q	-3	4	-7
R	-3	4	-7
S	-3	4	-7

O diagrama será então:



Note como o contexto G foi fechado e o contexto I foi aberto. Isto acontece no menor nível de detalhe (maior valor de $K = -3$). Se o nível de detalhe for incrementado, os conteúdos de F e G também serão visualizados.

Considerando que todos os nós já foram acessados, ou seja, que todos terão a sua função DOI calculada e fazendo o foco em G, calcula-se as funções de grau de interesse:

Nó	API	Dc	De	D	DOI
A	0	2		2	-2
B	-1	3		3	-4
C	-2	4		4	-6
D	-3	5		5	-8
E	-4	6		6	-10
F	-1	1		1	-2
G	-2	0	0	0	-2
H	-1	3	2	2	-3
I	-1	3	2	2	-3
J	-2	4	3	3	-5
K	-2	5	4	4	-6
L	-2	5	4	4	-6
M	-1	3		3	-4
N	-2	4		4	-6
O	-3	5		5	-8
P	-4	6		6	-10
Q	-3	1		1	-4
R	-3	1		1	-4
S	-3	1		1	-4

Para garantir os requisitos da Seção 4.2.1.2 que todos os browsers de hiperbase devem satisfazer, o maior valor de $K = -4$.

Para $DOI \geq -4$, tem-se o seguinte diagrama:

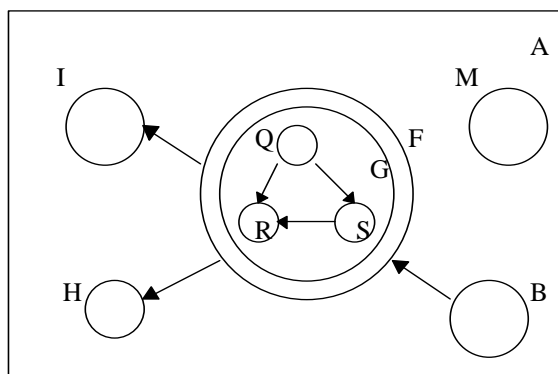


Figura 41: Foco em G e $DOI \geq -4$, apenas os nós diretamente conectados e os componentes de G são exibidos.

Para $DOI \geq -5$, outros nós já aparecem:

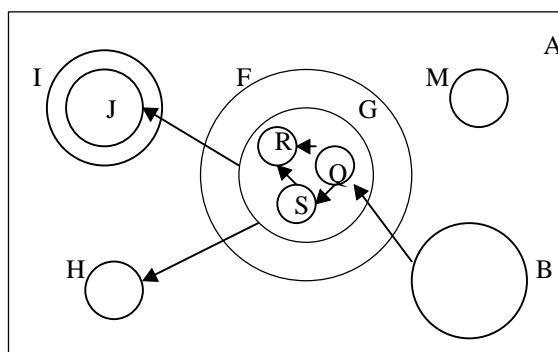


Figura 42: Foco em G e $DOI \geq -5$

Fazendo $DOI \geq -7$, temos:

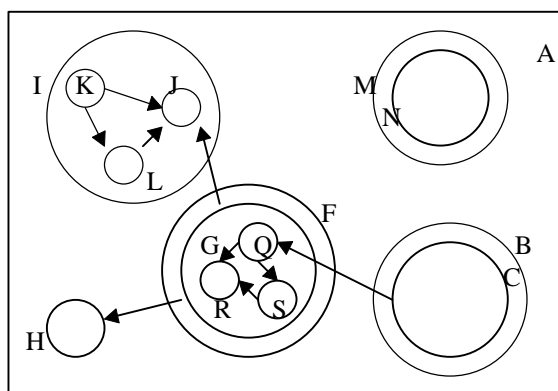


Figura 43: Foco em G e $DOI \geq -7$

Considerando, agora, que o nó J é o nó em foco, tem-se os seguintes valores para a função DOI(x,J):

Nó	API	Dc	De	D	DOI
A	0	2		2	-2
B	-1	3		3	-4
C	-2	4		4	-6
D	-3	5		5	-8
E	-4	6		6	-10
F	-1	3	2	2	-3
G	-2	4	3	3	-5
H	-1	3	5	3	-4
I	-1	1		1	-2
J	-2	0	0	0	-2
K	-2	2	1	1	-3
L	-2	2	1	1	-3
M	-1	3		3	-4
N	-2	4		4	-6
O	-3	5		5	-8
P	-4	6		6	-10
Q	-3	5		5	-8
R	-3	5		5	-8
S	-3	5		5	-8

Considerando DOI \geq -5, temos:

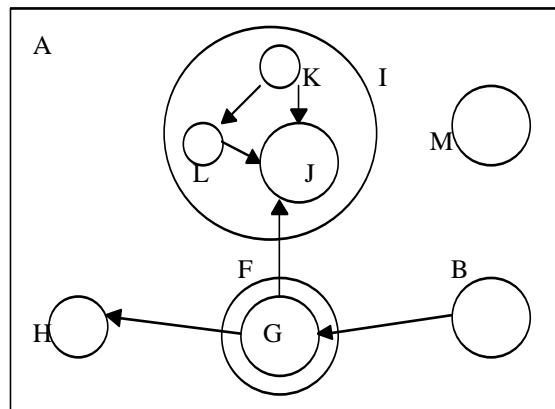


Figura 44: Foco em J e DOI \geq -5

Deve-se observar como alguns contextos são fechados quando muda-se o nó em foco de F para J, mas permanecendo no mesmo grau de interesse. Comparando as Figuras 42 e 44, verifica-se que os nó de contexto G foi fechado, omitindo seus componentes Q, R e S, e os nós K e L foram exibidos.

4.2.2 Navegação por Trilhas

Além da navegação por browsers, temos, no sistema HyperProp, um outro tipo de navegação, baseado no conceito de trilhas.

Trilhas são, na verdade, uma especialização de nós de composição que contém uma lista ordenada de nós e, recursivamente, trilhas.

O nó trilha possui um atributo especial que indica a que outra composição está associado, por exemplo, a um nó de contexto do usuário, a uma base privada ou à hiperbase pública. Este atributo obriga que todos os componentes da lista ordenada de nós estejam recursivamente contidos na composição a que a trilha está associada [SoCR95].

Outro atributo importante e específico das trilhas é o atributo visão. Este atributo associa cada nó da lista ordenada a uma perspectiva, possibilitando que duas ou mais perspectivas diferentes de um mesmo nó possam ser registradas na trilha.

A lista ordenada de nós reflete um caminho percorrido pelo usuário, que pode representar uma navegação durante uma sessão de trabalho ou sugerir uma ordem linear de leitura para o hiperdocumento.

Se a trilha registrar o caminho percorrido, enquanto o usuário navega pelo documento, cada nó acessado é colocado no fim da trilha, sendo que o mesmo nó pode ser inserido em mais de uma posição. A trilha que registra a trajetória durante uma sessão de navegação é chamada de *system private base trail*. Esta trilha é especial e gerenciada pelo sistema.

Caso a trilha sugira uma ordem linear de leitura do documento, ela deve ser previamente construída pelo autor, que deverá inserir nós (perspectivas) na lista de acordo com a ordem em que estes devem ser acessados.

As trilhas no Modelo de Contextos Aninhados podem estar armazenadas em bases privadas ou em um repositório de trilhas gerenciado pelo sistema.

As trilhas armazenadas em bases privadas estão associadas a nós componentes dessas bases e é responsabilidade do sistema garantir a consistência dessas trilhas. Se algum nó componente da trilha for eliminado da base, esta trilha deve ser imediatamente atualizada. Caso os componentes de uma base privada sejam transferidos para a hiperbase pública, as trilhas associadas a esses nós são transferidas para o repositório de trilhas.

No caso de uma operação de *shift* na base privada, onde todos os seus componentes são transferidos para a hiperbase pública, o sistema questionará se o usuário deseja salvar a trilha com a última navegação (*system private base trail*). Se a resposta for positiva, esta trilha será salva no repositório de trilhas estando associada não mais à base privada, mas sim à hiperbase pública, que é a composição que contém seus componentes.

No caso de uma operação de *check-out* de um nó de contexto de usuário da base privada, todas as trilhas associadas a este nó e seus componentes são transferidas para o repositório de trilhas.

Quando a navegação por trilhas é ativada, são fornecidos os seguintes comandos para o usuário:

- próximo: exibe o próximo nó da trilha;
- anterior: exibe o nó anterior ao corrente na trilha;
- primeiro: exibe o primeiro nó da trilha.

Além da navegação por comandos, estará disponível um browser de trilha, que terá praticamente as mesmas funções que um browser de contexto, exibindo a representação gráfica da trilha. Pelo browser será possível editar a trilha, acrescentando ou retirando nós e acessar qualquer um deles, em qualquer ordem.

No browser de trilha, são desenhados elos para indicar a ordem dos nós, porém na estrutura de dados do modelo conceitual estes elos não são necessários, já que a lista de nós componentes já reflete a ordem linear de navegação. Por isso, não faz sentido o usuário editar elos no browser de trilha.

4.3 Comparação com Trabalhos Relacionados

Para realizar a proposta de navegação por browsers e trilhas para sistemas hipermídia que permitem composições aninhadas de nós, foram examinados diversos sistemas hipermídia atuais e o que eles oferecem em relação a recursos gráficos para edição da rede e ferramentas de auxílio à navegação pelos nós e elos do documento. Assimilando a experiência de utilização dos vários sistemas estudados, o mecanismo de browsers e trilhas proposto visa agrupar suas melhores características.

Quando o usuário muda o nó em foco no browser de base, o diagrama é redesenhado, pois uma nova função de grau de interesse é calculada para cada nó, afetando quais nós e elos serão exibidos. Esta ação pode ser comparada ao que acontece no Thoth-II [UtYa89] onde o diagrama é atualizado dinamicamente à medida que o usuário percorre os nós do hipertexto. A diferença básica é que na proposta aqui apresentada, pode-se abrir e fechar contextos, mantendo a legibilidade do mapa para redes mais extensas, e no caso do Thoth-II, o tamanho da rede representa um grande problema.

Nos browsers propostos, garante-se que todos os nós diretamente relacionados ao nó em foco estejam sempre visíveis no diagrama, exercendo a funcionalidade do mapa do Electronic Document System [YaMe85] que exibe o nó corrente no centro da tela e todos os possíveis nós de onde o leitor pode ter vindo e para onde ele pode ir.

No caso do Intermedia [UtYa89], a tentativa de construir um mapa global não foi bem sucedida, porque o modelo de dados não possui nenhuma forma de hierarquizar ou agrupar informações. Os modelos hipermídia que suportam composições aninhadas de nós permitem a

construção de uma rede de nós bem estruturada, agrupando partes da hierarquia em nós de composição. Isto viabiliza a construção de mapas globais que utilizem apenas um ícone para representar vários nós da rede, diminuindo o problema do tamanho do diagrama, similarmente ao que é sugerido pelo CYBERMAP. Para melhorar a legibilidade do mapa, utiliza-se a estratégia da visão olho-de-peixe, tal como já foi experimentado pelo Intermedia, permitindo ainda que o usuário defina alguns nós como pontos de referência, sendo que estes nós serão sempre exibidos no diagrama. Além disso, os browsers são atualizados dinamicamente quando o nó em foco é modificado ou quando ocorre alguma alteração na estrutura de contextos, o que garante a fidelidade do diagrama em relação ao conteúdo do documento. Tal qual o Intermedia, a navegação é permitida, oferecendo ao usuário a opção de acessar diretamente qualquer nó sendo exibido. Apesar do browser oferecer um layout automático do diagrama, o usuário também poderá modificar a posição dos nós, se desejar, fazendo com que o mapa possa ser customizado por cada usuário separadamente. O problema da discordância entre os usuários sobre o layout final pode ser solucionado pela criação de um perfil para cada usuário, que indicaria como o diagrama deveria ser visualizado. A construção deste perfil ainda não foi estudada, mas representa uma boa proposta para trabalhos futuros.

O CYBERMAP [Gloo91] sugere agrupar nós relacionados representando-os com apenas um ícone no diagrama (HYPERDRAWER), porém isto exige um esforço computacional prévio para identificar a semelhança entre os nós. Em modelos hipermídia que suportam composições aninhadas de nós, este esforço é dispensado, pois a própria hierarquia de contextos já fornece a semelhança entre os nós. Idealmente, os nós componentes de um contexto devem ter bastante semelhança, permitindo que o contexto-pai represente todos os seus componentes no browser seguindo a filosofia do HYPERDRAWER do CYBERMAP.

Os browsers do Storyspace são ferramentas importantes para a edição e navegação dentro da estrutura dos hiperdocumentos, porém, para garantir a legibilidade do diagrama, o sistema limita o número de níveis de hierarquia que podem ser visualizados (apenas dois níveis), não oferecendo uma ferramenta muito flexível onde o usuário ajusta o nível de detalhe de informações que deseja visualizar. A proposta apresentada, além de oferecer a possibilidade de ajuste do nível de detalhe do diagrama, utiliza um mecanismo de filtragem de informações mais inteligente, considerando a hierarquia e as relações entre os nós baseando-se no modelo de visões olho-de-peixe.

Deve-se notar que a implementação de browsers e editores no Sistema HyperProp é feita através de métodos associados aos nós e não como novas entidades da estrutura de dados, como por exemplo, no NoteCards [HaMT87] [Hala88].

Para o Modelo de Contextos Aninhados, não é necessário ter um mecanismo de trilhas extremamente poderoso como o do sistema Scripted Documents [Zell89], pois o próprio modelo conceitual já o é. No Scripted Documents, o único mecanismo de ligação é através de trilhas, por isso é bem abrangente. Mesmo assim, o mecanismo de trilhas do MCA permite trilhas seqüenciais, pela própria definição de trilha; trilhas procedurais, pois é possível aninhar trilhas; e trilhas variáveis, por exemplo, onde o próximo nó da trilha é a versão corrente ou qualquer outro tipo de query. A trilha branching [Zell89] pode ser feita pelo próprio conceito de nós de contexto e elos, porque eles já representam uma sub-rede do hipertexto.

A trilha que registra o caminho percorrido pelo usuário também é oferecida nos sistemas Neptune, Electronic Document System, NoteCards (History List), Intermedia, Hypercard e no Storyspace. No caso do Hypercard, a ferramenta Recent [Will87] não reflete exatamente o caminho percorrido pelo usuário porque não permite repetição de nós na trilha. No Modelo de Contextos Aninhados, estas repetições são permitidas e o mesmo nó será exibido na trilha quantas vezes for acessado pelo usuário considerando a perspectiva de exibição. Além disso, o próprio autor pode definir trilhas sobre o documento, obtendo a mesma funcionalidade do Guided Tours [MaIr89], implementado sobre o NoteCards. Estas trilhas definem “viagens” pela rede que podem ser percorridas passo a passo ou navegadas/editadas pelo browser de trilha.

A proposta do algoritmo para cálculo da função de grau de interesse dos nós nas visões olho-de-peixe apresentada nesta dissertação é uma contribuição relevante ao modelo inicial da Furnas. Para o cálculo da distância dos nós em relação ao nó em foco, que representa a posição do usuário no espaço de navegação, utilizamos métricas sensíveis tanto a navegação em profundidade quanto à navegação por elos. Desta forma, o grau de interesse dos nós reflete sua distância abstrata ao nó corrente (localização do usuário no hiperdocumento). Esta proposta difere das extensões que foram apresentadas em [SaBr92] e [Noik93], descritas nas Seções 3.5.1 e 3.5.2 respectivamente. Estes outros autores introduzem modificações no layout do grafo, alterando o tamanho e a posição dos nós de acordo com seu grau de interesse. Isto poderia ser acrescentado à proposta desta dissertação para realizar o layout final da estrutura do hiperdocumento.

Após apresentar a proposta de construção de browsers e trilhas para documentos hipermídia com composições aninhadas, discute-se a implementação destas ferramentas sobre o sistema HyperProp, o que será abordado no Capítulo 5.