

Capítulo 5

Implementação

Com o propósito de validar a proposta de construção de browsers e mecanismo de trilhas apresentada, foi feita uma implementação sobre o sistema hipermídia HyperProp, baseado no Modelo de Contextos Aninhados.

O sistema HyperProp está sendo desenvolvido com o objetivo de fornecer um conjunto de ferramentas completo para manipulação de documentos hipermídia em ambiente multiplataforma. A implementação atual já representa o início para construção do sistema utilizando uma arquitetura distribuída, sofrendo algumas mudanças em relação à primeira implementação [Bati94].

Na proposta de arquitetura distribuída para o sistema HyperProp, o subsistema de armazenamento, responsável pela persistência dos objetos, é parte do módulo servidor e o subsistema de apresentação é integrante do módulo cliente (vide Figura 45) [SoCC93]. As ferramentas de browsing e o mecanismo de navegação por trilhas fazem parte do módulo cliente do sistema hipermídia HyperProp, com exceção do browser de hiperbase, que é um método da hiperbase pública, gerenciada pelo módulo servidor.

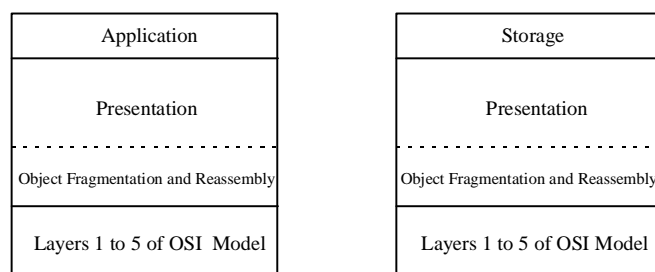


Figura 45: HyperProp - arquitetura do cliente (esquerda) e servidor (direita)

Neste capítulo será apresentado um breve relatório sobre a implementação das ferramentas de auxílio à navegação, ressaltando algumas características e funções oferecidas pelos browsers e mecanismo de trilhas do sistema HyperProp. Na Seção 5.1 fala-se sobre a escolha do pacote gráfico para tratamento do layout automático de grafos. Na Seção 5.2 discute-se a implementação do browser de hiperbase, reservando para a Seção 5.3, a discussão sobre a implementação do browser de base e para a Seção 5.4, a discussão sobre o browser de contexto. Na Seção 5.5 será apresentada a implementação do mecanismo de navegação por trilhas.

5.1 Ferramentas para Layout Automático de Grafos

Apesar de ter sido feito um breve estudo na área de layout automático de grafos, com o objetivo de definir as técnicas que seriam utilizadas para apresentar o layout dos diagramas estruturais de nós e elos nos browsers gráficos, não julgou-se de interesse imediato a implementação dos algoritmos para realizar tal função. Acredita-se que esta tarefa é bastante complexa e merece um estudo bem mais aprofundado, fugindo do objetivo central dessa dissertação. Assim sendo, desde o início do trabalho, pensou-se em utilizar alguma ferramenta ou pacote já implementado que disponibilizasse tais funções.

Na procura por um pacote gráfico que oferecesse os serviços de layout automático de grafos, foram pesquisados alguns sistemas desenvolvidos.

O primeiro sistema analisado foi o EDGE [PaTi90]. O EDGE é um núcleo editor para manipulação visual e direta de grafos que pode ser adaptado rapidamente para diversas aplicações específicas. Sua proposta parecia razoável para a construção dos browsers, pois o editor oferecia layout automático de grafos, possibilidade de definir abstrações através de subgrafos, adaptabilidade a várias aplicações com código fonte em C++ e também persistência de grafos. O único detalhe é que os autores não estavam mais trabalhando na ferramenta, logo não existia mais suporte, inclusive para adaptá-lo a novas versões do sistema operacional Sun-OS, que é a plataforma utilizada pelo EDGE.

O segundo sistema analisado foi o GraphEd [Hims95a] [Hims95b] sendo desenvolvido na Universidade de Passau na Alemanha. GraphEd também é um editor gráfico extensível seguindo a mesma idéia do EDGE. Entretanto, com uma diferença básica, não se concentra somente em editar ou visualizar grafos, mas oferece uma interface para a implementação de novos algoritmos para layout de grafos. GraphEd também parecia uma boa opção, porém houve dificuldades para conseguir instalar o software no ambiente operacional disponível.

Outra opção analisada foi o sistema *daVinci* [FrWe95], desenvolvido na Universidade de Bremen na Alemanha, que é apenas um visualizador de grafos, não oferecendo operações de edição. Logo, não seria o mais adequado para a construção dos browsers, ainda mais pelo fato de não disponibilizar seu código fonte e não oferecer tratamento para subgrafos aninhados.

Finalmente, a opção mais adequada que oferecia todas as funcionalidades procuradas, dando suporte à visualização e edição de grafos compostos, usando o algoritmo descrito em [SuMi91] e comentado na Seção 3.3, foi o sistema D-ABDUCTOR [Misu94]. D-ABDUCTOR é um sistema desenvolvido no Institute for Social Information Science, Fujitsu Laboratories Ltd. no Japão. Seu objetivo é suportar o processo de raciocínio dinâmico dos seres humanos através do uso de algoritmos sofisticados para desenho de grafos. Foi desenvolvido para integrar a capacidade de raciocínio humano e a capacidade de processamento de informações dos computadores, pois os diagramas são uma boa mídia para refletir e organizar pensamentos pessoais e se comunicar com outras pessoas em trabalhos cooperativos.

D-ABDUCTOR provê uma série de facilidades para lidar com diagramas, porém, a característica mais importante em relação a proposta apresentada nesta dissertação é o fato de representar tanto relações de adjacência como relações de inclusão entre os nós, se enquadrando perfeitamente no tipo de ferramenta que estava sendo procurada. D-ABDUCTOR foi o primeiro sistema que oferece tais facilidades, baseado no layout automático de grafos, e ainda oferece um recurso bastante interessante, que é mostrar as mudanças nos diagramas através de animação. Como o layout automático de grafos pode mudar drasticamente o desenho do diagrama, usando recursos de animação, consegue-se reduzir a mudança visual instantânea, preservando o mapa mental do usuário e ajudando sua orientação na estrutura apresentada.

O sistema D-ABDUCTOR não foi implementado com o objetivo de ser um editor extensível e facilmente adaptável a outras aplicações. Logo, os algoritmos de layout estão intimamente relacionados com a interface gráfica utilizada pelo sistema, que é o pacote X-View. Como o esforço para modularizar somente os algoritmos de layout automático de grafos do sistema seria muito grande, optou-se por construir a interface gráfica dos browsers e trilhas do sistema HyperProp utilizando o mesmo pacote gráfico. Desta forma, pôde-se aproveitar o código fonte para os algoritmos de layout sem muitas alterações. O código fonte do sistema está escrito em linguagem C, no padrão Kernighan & Ritchie, e pode ser compilado usando o compilador GNU C.

5.2 Browser de Hiperbase

O browser de hiperbase é o responsável por oferecer uma visão gráfica da estrutura de nós e elos da hiperbase pública do Modelo de Contextos Aninhados. A hiperbase pública é, por definição, um repositório das informações consistentes do sistema HyperProp, ou seja, é um conjunto de todos os nós terminais e de contexto de usuário que podem ser acessados por usuários, desde que estes tenham os devidos direitos de acesso.

Quando o browser de hiperbase é ativado, ele oferece uma visão de todos os nós a que o usuário tem acesso (vide Figura 46). Na implementação corrente, o browser exibe todos os nós da hiperbase pública, pois o mecanismo de controle de acessos ainda não está implementado. Os nós são representados por retângulos no diagrama, sendo que os nós terminais e nós de contexto de usuário são desenhados com cores diferentes (verde para nó terminal e azul para nó de contexto de usuário).

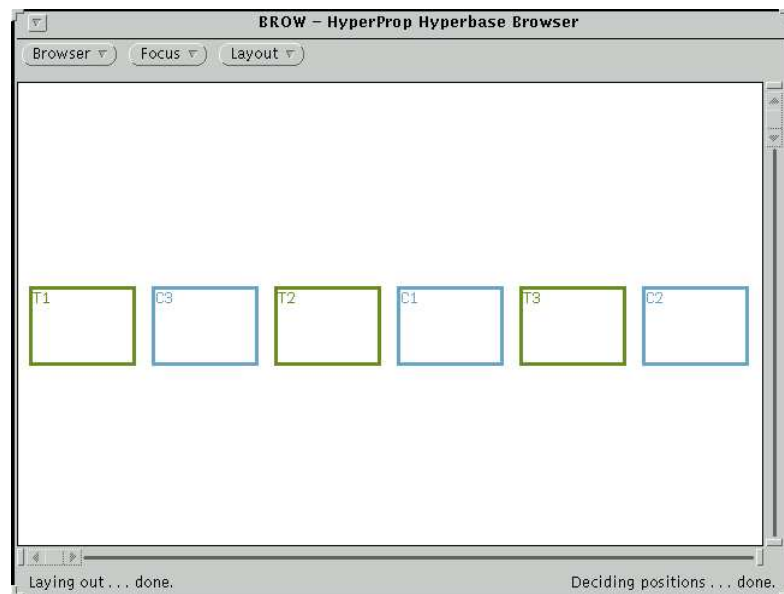


Figura 46: HyperProp - primeira visão do browser de hiperbase

Quando o usuário aciona o botão esquerdo do mouse duas vezes sobre um nó de contexto de usuário específico, o diagrama exibe os nós componentes e elos deste nó. Esta é a primeira visão de um nó de contexto de usuário no browser de hiperbase (vide Figura 47). A partir daí, quando o usuário focalizar algum nó específico, o diagrama é atualizado de acordo com o algoritmo de visões olho-de-peixe, descrito na Seção 4.2.1.2. O usuário pode focalizar um nó específico, selecionando o nó com o mouse e escolhendo a opção “Focus Selected” do menu “Focus”, ou através do menu popup ativado com o botão direito do mouse. O nó focalizado é desenhado com marcas nos vértices do retângulo que o identificam como foco atual de interesse do usuário (vide Figura 48). O usuário pode mudar o nível de detalhe exibido no diagrama escolhendo a opção “Detail” do menu “Layout”, onde é possível escolher, sob a forma de porcentagem, a quantidade de informação exibida.

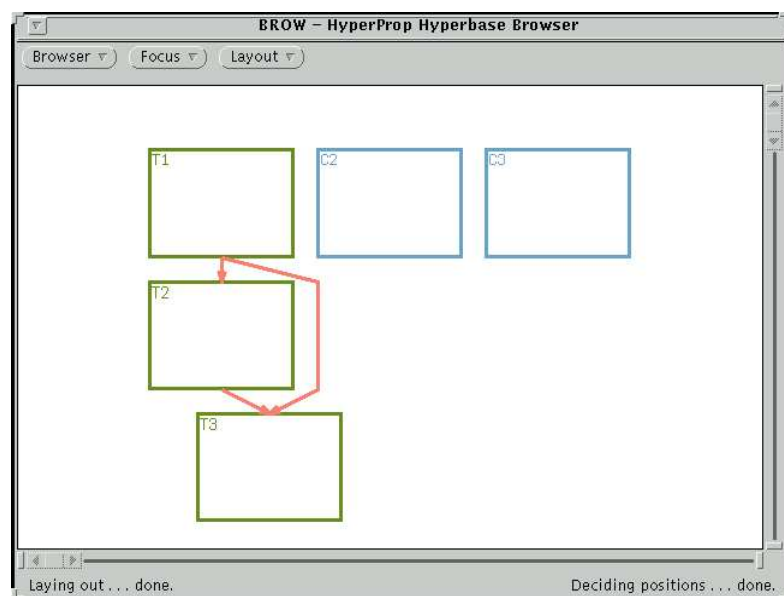


Figura 47: HyperProp - depois de escolher o contexto de usuário em questão no browser de hiperbase (contexto C1)

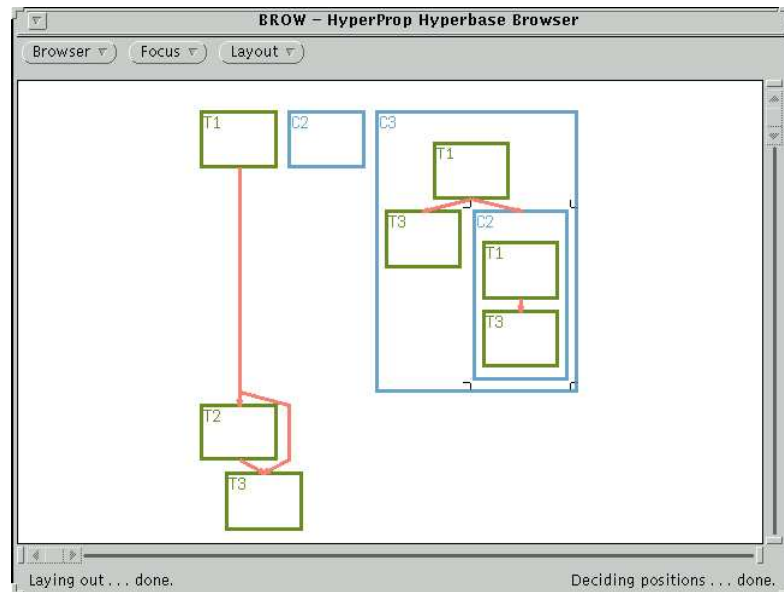


Figura 48: HyperProp - focalizando o nó C2 no browser de hiperbase

A função de grau de interesse é computada apenas para os componentes diretos do nó de contexto de usuário em questão e para os componentes dos nós já focalizados. A cada vez que o usuário muda o foco, novos nós também são considerados para o algoritmo do olho-de-peixe. Se o usuário não quiser mais que os componentes de um nó de contexto de usuário sejam levados em conta para o cálculo da função de grau de interesse, o usuário deve selecionar o nó de contexto de usuário e escolher a opção “Collapse Selected” do menu “Focus”, também disponível pelo menu popup ativado com o botão direito do mouse.

Para a definição de landmarks, o usuário deve selecionar um nó e acionar a opção “Landmark” do menu “Focus” ou ativar o menu popup com o botão direito do mouse. Esta opção transforma o nó e sua perspectiva corrente em landmarks, caso eles não o sejam, ou tira a indicação de landmark do nó e seus componentes recursivos, caso eles já o sejam. Os landmarks são desenhados com linhas mais espessas no diagrama, para diferenciá-los dos nós comuns (vide Figura 49).

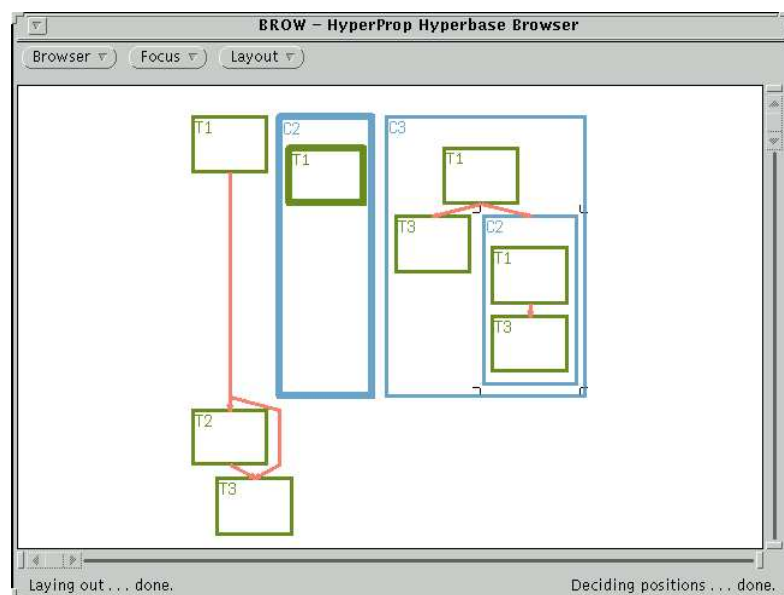


Figura 49: HyperProp - exemplo de landmarks no browser de hiperbase

O browser oferece ainda a possibilidade de mudar algumas características dos elementos do diagrama, tais como forma e cor, o que está disponível através da opção “Elements” do menu “Layout”. O usuário também tem a opção de mudar a direção do layout automático do diagrama (norte, sul, leste e oeste) (vide Figura 50) através da opção “Options” do menu “Layout”. Pode também mudar a escala do diagrama aproximando ou afastando o desenho (efeito zoom) através da opção “Scale” do menu “Layout”.

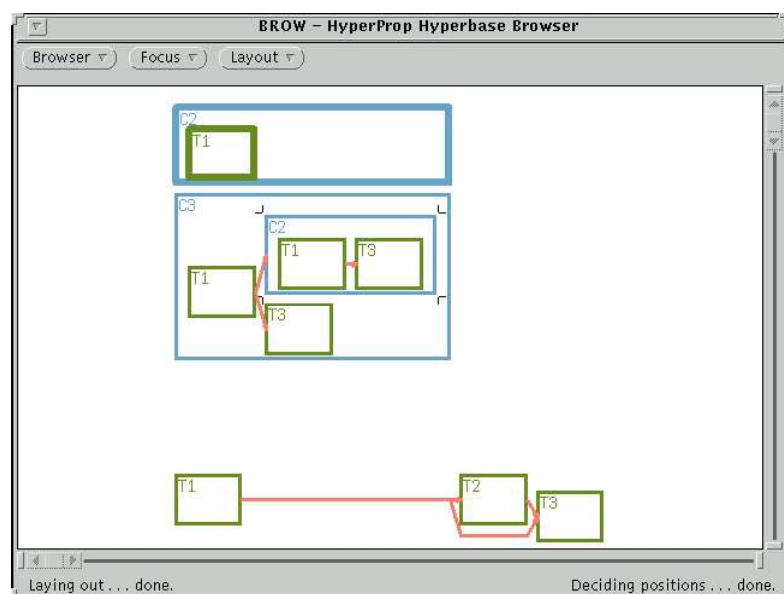


Figura 50: HyperProp - mesmo layout da Figura 49 na direção leste

Caso o usuário queira voltar à visão inicial do browser de hiperbase, onde todos os nós são desenhados no diagrama, deverá escolher a opção “Initialize” do menu “Browser”. Esta opção reinicializará o conteúdo do diagrama, relendo o conteúdo da hiperbase pública e apresentando uma visão atualizada do repositório de nós.

Caso o usuário queira sair da ferramenta de visualização, deve fazê-lo através da opção “Quit” do menu “Browser”.

O browser de hiperbase foi implementado utilizando o algoritmo proposto na Seção 4.2.1.2 para filtrar as informações exibidas ao usuário, oferecendo a possibilidade de adequar o nível de detalhe do mapa à sua necessidade.

O algoritmo utilizado para realizar o layout automático da estrutura em grafo da rede de nós e elos foi o apresentado em [SuMi91], cedido pela Fujitsu Laboratories Ltd. através do código fonte do sistema D-ABDUCTOR [Misu94].

A interface gráfica foi construída utilizando o pacote X-View, disponível nas plataformas Sun-OS, Solaris e Linux. Os algoritmos para cálculo das visões olho-de-peixe foram escritos na linguagem C++ [Schi90] [Schi92], usando o compilador GNU C++ (vide Anexo 1).

Para a implementação das visões olho-de-peixe, foi necessária a definição de uma estrutura que relaciona nós do Modelo de Contextos Aninhados a vértices exibidos no diagrama estrutural, já que a estrutura de dados utilizada pelo algoritmo de layout de grafos no sistema D-ABDUCTOR foi preservada. Essa estrutura chama-se *node_perspective* e é utilizada para computar as visões olho-de-peixe.

Cada nó/perspectiva no modelo conceitual é representado por um vértice no diagrama estrutural, logo, cada nó está replicado na estrutura para cálculo das visões olho-de-peixe de acordo com o número de perspectivas que possui.

Os elos definidos nas composições do Modelo de Contextos Aninhados não são replicados na estrutura *node_perspective*. Para calcular a distância em elos, componente da função de grau de interesse da estratégia olho-de-peixe, são utilizadas funções da implementação do Modelo de Contextos Aninhados para recuperação de elos.

Para maiores detalhes sobre a implementação do algoritmo para computar a função de grau de interesse da estratégia olho-de-peixe, deve-se consultar o Anexo 1, que exhibe o código fonte do programa.

O browser de hiperbase é um método da hiperbase pública no sistema HyperProp e permite que os usuários visualizem sua estrutura de nós e elos em vários níveis de detalhe.

5.3 Browser de Base

O browser de base tem como função básica oferecer uma interface gráfica para manipulação do conteúdo das bases privadas do sistema HyperProp. As bases privadas representam as sessões de trabalho dos usuários.

Esta ferramenta apresenta todas as funções descritas no browser de hiperbase, porém com algumas funcionalidades a mais. O browser oferece todas as operações de edição e navegação disponíveis em uma base privada. O usuário pode criar uma nova base privada ou abrir outra existente através da opção “New/Load” do menu “Browser”. A criação de novos nós de contexto de usuário e nós terminais é feita através do menu “Insert”. A eliminação destes nós é feita através do menu “Edit”, opção “Delete”. O menu “Layout” oferece as mesmas opções que existem no browser de hiperbase, permitindo que se modifique o nível de detalhe do diagrama, se mude a apresentação dos elementos, se modifique o layout automático e a escala do mapa. O menu “Focus” também tem as mesmas funcionalidades do browser de hiperbase, onde o usuário pode focalizar um novo nó (opção “Focus Selected”), esconder o conteúdo de um nó de contexto de usuário do diagrama (opção “Collapse Selected”) e definir nós importantes através de landmarks (opção “Landmark”). A interface gráfica do browser de base é apresentada na Figura 51.

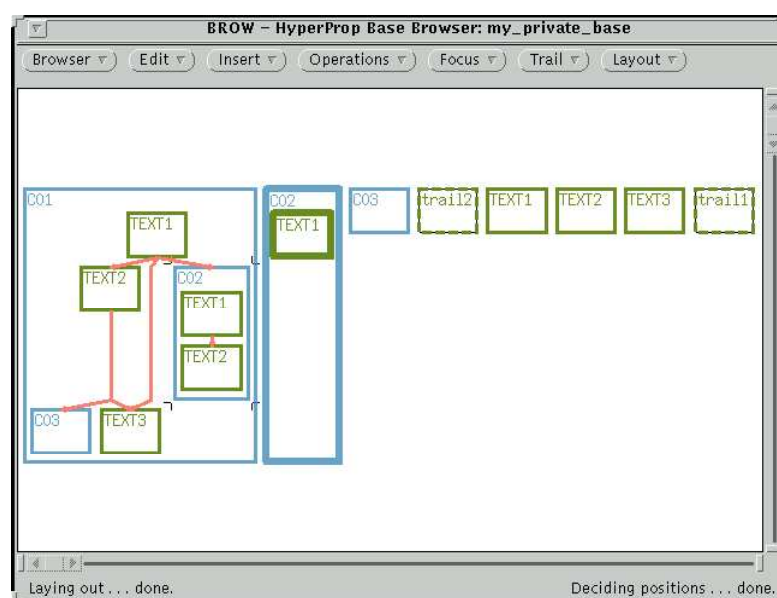


Figura 51: HyperProp - browser de base

A base privada oferece ainda operações para trazer nós da hiperbase pública ou mover nós para este repositório. Estas operações estão disponíveis no menu “Operations”. A opção “Check-out” move o nó selecionado para a hiperbase pública. A opção “Shift” move todo o conteúdo da base privada (nós terminais e de contexto de usuário) para a hiperbase pública. As opções “Check-in” e “Open” trazem nós da hiperbase pública de acordo com os procedimentos de *check-in* e *open* descritos na Seção 4.2.1.3. Estas opções abrem uma janela com uma lista de todos os nós da hiperbase pública a que o usuário tem direito de acesso, separados por tipo (terminal ou contexto de usuário) (vide Figura 52). Em implementações futuras, será oferecida a possibilidade de transferir nós das bases privadas para a hiperbase pública e vice-versa, através de operações de *drag & drop* entre os browsers de base e de hiperbase.

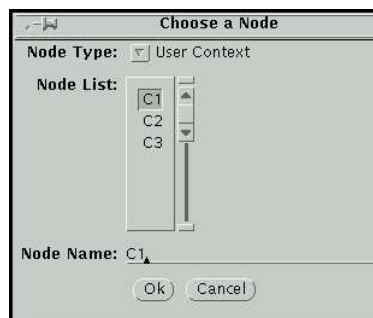


Figura 52: HyperProp - janela para escolha do nó a ser dado *open* ou *check-in*

No browser de base, o usuário pode fazer as composições simplesmente movendo os nós componentes para dentro do contexto de usuário desejado, desde que este contexto esteja sendo exibido no browser. A edição de elos também é permitida, basta acionar o botão esquerdo do mouse sobre o nó origem e pressionando a tecla <SHIFT> levá-lo até o nó destino. Após a definição das extremidades do elo, o sistema verifica quais os nós de contexto comuns nas perspectivas dos dois nós e fornece uma lista com todos estes nós para que o usuário selecione em que contexto será inserido o elo. Vale ressaltar que um elo deve ser definido dentro de um contexto componente da base privada, pois não existem elos definidos na própria base privada, a não ser os das anotações, que ainda não foram implementadas. Na implementação corrente, apenas os elos “um para um” estão sendo tratados, os elos “m para n” serão implementados futuramente.

O algoritmo de layout automático de grafos e a interface gráfica do browser de base seguem os mesmos padrões utilizados no browser de hiperbase, descritos na seção anterior.

Para realizar a transferência de informações entre as bases privadas e a hiperbase pública, definiu-se os métodos da classe base privada para realizar tais funções:

- `void CheckOut (Node *node);`

utilizada para executar a operação de *check-out* em um nó de uma base privada, onde identifica-se o nó que deve ser movido para a hiperbase pública.

- `void Shift ();`

utilizada para executar a operação de *shift* em uma base privada que terá seu conteúdo transferido para a hiperbase pública.

- `Node * CheckIn (Node *node, char *version_name);`

utilizada para executar a operação de *check-in* na base privada, onde identifica-se o nó da hiperbase pública e o nome da nova versão deste nó na base privada, sendo que a função retorna a nova versão criada.

- `Node * Open (Node *node, char *version_name);`

utilizada para executar a operação de *open* na base privada, onde identifica-se o nó da hiperbase pública e o nome da nova versão deste nó na base privada, sendo que a função retorna a nova versão criada.

Para maiores detalhes sobre a implementação destas funções deve-se consultar o Anexo 2.

O browser de base é um método das bases privadas do sistema HyperProp e permite que os usuários visualizem e manipulem a estrutura de nós e elos das bases privadas apresentando o diagrama em vários níveis de detalhe.

5.4 Browser de Contexto

O browser de contexto é uma ferramenta para edição gráfica do conteúdo de um nó de contexto de usuário específico, por isso possui uma filosofia bem mais simples que os outros browsers apresentados. Esta ferramenta exibe apenas os componentes diretos do nó de contexto de usuário desejado, não utilizando filtros para esconder informações.

O browser de contexto é ativado quando o usuário aciona duas vezes o botão esquerdo do mouse sobre um nó de contexto de usuário em uma base privada. Na ferramenta, é possível criar/abrir novos nós de contexto de usuário através da opção “New/Load” do menu “Browser”. O usuário insere novos nós terminais ou de contexto do usuário através do menu “Insert” ou elimina nós indesejados através da opção “Delete” do menu “Edit”. O menu “Layout” apresenta funções já descritas nos outros browsers para editar características do layout automático do diagrama. Nesta ferramenta o usuário não pode mudar o nível de detalhe apresentado, pois o diagrama exibe apenas os componentes diretos do nó de contexto em questão (vide Figura 53).

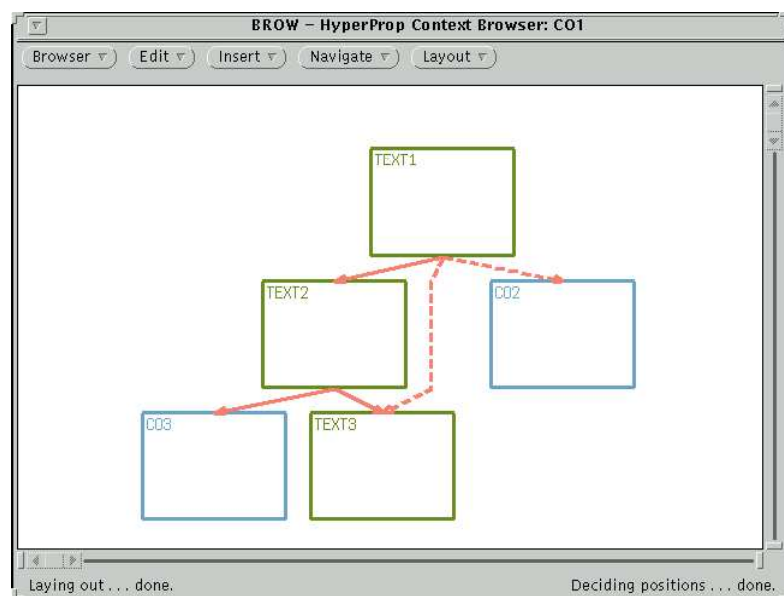


Figura 53: HyperProp - browser de contexto

A criação de elos no browser de contexto pode ser feita de duas maneiras. A mais simples é desenhando o elo, com o mouse, do nó origem para o nó destino, basta acionar o botão esquerdo do mouse no nó origem, teclando <SHIFT>, e com o botão do mouse pressionado levá-lo até o nó destino. Esta maneira de criar elos exige que os nós origem e destino sejam componentes diretos do nó de contexto de usuário em questão. Se o usuário desejar criar elos que envolvam componentes recursivos do nó de contexto de usuário em questão, ele deve escolher a opção “Link” do menu “Insert”. Para escolher nós internos aos componentes do nó de contexto, ele deve navegar em profundidade pelos contextos até que atinja o nó terminal extremidade (origem ou destino) final do elo. Os elos “m para n” não são tratados na implementação corrente, mas serão manipulados em trabalhos futuros.

Para inserir outros nós da base privada no nó de contexto de usuário sendo editado, basta acionar a opção “Private Base Node” do menu “Insert”. O sistema apresentará uma lista com todos os nós da base privada não componentes do contexto em questão, separados por tipo (terminal ou contexto de usuário), para que o usuário escolha qual deles será inserido. Em implementações futuras, será oferecida a possibilidade de inserir nós da base privada no contexto sendo editado através de operações de *drag & drop* entre os browsers de base e de contexto.

A navegação em profundidade pela perspectiva corrente é feita acionando o botão esquerdo do mouse duas vezes no nó de contexto de usuário desejado, ou através do menu “Navigate”, opção “Go down”. Caso o usuário deseje retornar aos nós mais externos da perspectiva corrente, basta selecionar a opção “Go up” do menu “Navigate”.

A interface gráfica do browser de contexto e o algoritmo para layout automático de grafos seguem os mesmos padrões descritos nos outros browsers (vide Seção 5.1).

O browser de contexto é um método dos nós de contexto do sistema HyperProp e permite que se visualize e manipule sua estrutura de nós e elos.

Assim que o mecanismo de controle de versões estiver totalmente implementado na versão atual do sistema HyperProp, este mesmo browser de contexto também será utilizado para visualizar o conteúdo dos nós de contexto de versões.

5.5 Mecanismo de Navegação por Trilhas

O mecanismo de navegação por trilhas do sistema HyperProp tem como objetivo fornecer aos usuários a possibilidade de navegar em uma trajetória percorrida durante uma sessão anterior de navegação e de criar, manipular e manter caminhos pré-definidos sobre um determinado documento.

As trilhas no Modelo de Contextos Aninhados são especializações dos nós de composição, e podem conter nós terminais, nós de contexto de usuário e trilhas, recursivamente. Cada nó trilha tem um atributo que indica a que nó de contexto está associada e outro atributo que indica a perspectiva de cada ocorrência de um nó pertencente à

trilha, ressaltando que um mesmo nó pode ser incluído mais de uma vez na trilha, sob perspectivas diferentes ou sob a mesma perspectiva.

As trilhas podem estar armazenadas em bases privadas ou em um repositório de trilhas gerenciado pelo sistema. As trilhas do repositório de trilhas do sistema estarão sempre associadas à hiperbase pública ou a seus nós de contexto de usuário componentes e não podem ser editadas.

Quando o usuário ativa o browser de base, a trilha que registrará o seu caminho durante a sessão de trabalho é ativada. Esta trilha é gerenciada pelo sistema e é chamada de *system private base trail*. Caso o usuário deseje ativar uma outra trilha para definir um caminho sobre um documento, deve selecionar a trilha com o mouse e ativar a opção “Record Trail”. Se o objetivo for navegar por uma trilha existente, o usuário deve acionar o botão esquerdo do mouse duas vezes sobre a trilha e ativar o browser de trilha. Mesmo com outra trilha ativa, a *system private base trail* continua sendo atualizada com a navegação do usuário.

Garantindo a consistência das trilhas, só são inseridos nós na trilha ativada se estes nós forem componentes recursivos do nó de contexto a que a trilha está associada. Se o usuário acessar um nó que não seja componente do nó associado da trilha, a definição da trilha é imediatamente interrompida. Uma outra maneira de parar a definição de uma trilha é através da opção “Stop Trail” do menu “Trail” do browser de base.

Para visualizar e editar a trajetória percorrida durante a sessão de navegação corrente, deve-se ativar a opção “System Trail” do menu “Trail” do browser de base, que ativa o browser de trilha da *system private base trail*. Esta trilha é sempre atualizada com a última navegação do usuário, mesmo que o usuário esteja navegando por seus nós componentes.

Toda vez que o usuário realiza uma operação de *shift* na base privada, o sistema salva todas as trilhas componentes da base privada no repositório de trilhas e questiona se ele deseja salvar a *system private base trail* nesse repositório, pedindo o novo nome a ser utilizado. Quando esta operação é realizada, o nó associado a esta nova trilha, criada no repositório de trilhas, é agora a hiperbase pública. A *system private base trail* é então reinicializada.

Quando o usuário realiza uma operação de *check-out* em um nó de contexto da base privada, todas as trilhas associadas a este nó e a seus componentes recursivos são transferidos para o repositório de trilhas.

Para que o usuário possa manipular trilhas armazenadas no repositório de trilhas, ele deve trazer estas trilhas para uma base privada, através da opção “Get Trail” do menu “Trail” do browser de base. Esta opção traz uma cópia da trilha para a base privada e realiza uma operação de *check-in* no nó de contexto a que a trilha está associada e em cada um de seus componentes, na medida em que são acessados pela navegação na trilha. Assim, o usuário pode manipular a trilha e, se quiser, salvá-la no repositório de trilhas, através da opção “Save Trail”, ressaltando que ela é uma nova trilha e não tem nenhuma relação com a trilha que a originou. Quando o usuário salva uma trilha da base privada no repositório de trilhas, é

realizada uma operação de *check-out* automática no nó de contexto a que a trilha está associada.

O usuário pode criar uma nova trilha em uma base privada através do menu “Insert” do browser de base, acionando a opção “Trail Node”. Se o objetivo for editar e manipular as trilhas componentes das bases privadas, deve-se ativar o browser de trilha acionando o botão esquerdo do mouse duas vezes sobre a trilha desejada. Quando o usuário ativa um browser de trilha ou um browser de contexto durante uma sessão de trabalho, a *system private base trail* continua sendo atualizada à medida que o usuário acessa novos nós.

Com o browser de trilha, é possível visualizar os componentes da trilha, navegar por ela e editá-la incluindo ou excluindo componentes (vide Figura 54). Apesar de eles serem exibidos no browser, indicando a ordem dos nós, não é possível editá-los pois, na verdade, não existem elos na composição trilha, já que a própria lista de nós já reflete a ordem dos componentes.

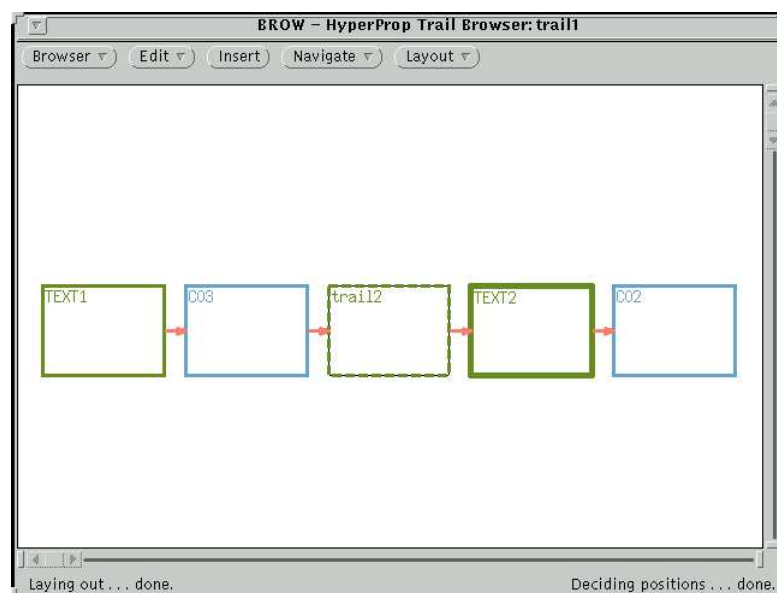


Figura 54: HyperProp - browser de trilha

O usuário pode criar uma nova trilha na base privada ou abrir outra já existente através da opção “New/Load” do menu “Browser”. Quando se cria uma nova trilha, deve-se dizer qual o nó de contexto associado. Para isso, o sistema fornece uma lista com todos os nós de contexto componentes da base privada onde a trilha será inserida.

O usuário pode excluir nós da trilha através da opção “Delete” do menu “Edit”, depois de selecioná-los com o mouse. A opção de incluir novos nós na trilha é dada pelo menu “Insert”. Esta opção abre uma janela para seleção dos nós recursivamente contidos no nó de contexto a que a trilha está associada, ou de trilhas associadas a estes nós. Após a escolha do usuário, o nó é inserido após o nó corrente da trilha. Em implementações futuras, será oferecida a possibilidade de inserir novos nós na trilha através de operações de *drag & drop* entre o browser de trilha e o browser de contexto ou o browser de base, dependendo se a trilha está associada a um nó de contexto de usuário ou a uma base privada.

O sistema garante a consistência de todas as trilhas, ou seja, caso algum nó componente do contexto a que a trilha está associada seja excluído deste contexto, exclui-se imediatamente este nó da trilha, pois todos os componentes de uma trilha têm que estar necessariamente contidos no nó de contexto associado ou em seus componentes, recursivamente.

O nó corrente da trilha é desenhado com bordas mais grossas para diferenciá-lo dos outros nós. As trilhas componentes da trilha sendo manipulada são desenhadas com bordas tracejadas, para diferenciá-las dos outros nós terminais e de contexto componentes.

O menu “Navigate” oferece as operações de navegação na trilha. São elas:

- *Home*: vai para o primeiro nó da trilha;
- *Next*: vai para o próximo nó da trilha;
- *Previous*: vai para o nó anterior ao corrente da trilha;
- *Go up* e *Go down*: navega em profundidade pelas trilhas componentes da trilha.

O menu “Layout” tem a mesma funcionalidade deste mesmo menu no browser de contexto, podendo-se modificar a aparência dos nós no diagrama, mudar a direção do layout e a escala do mapa.

A interface gráfica do browser de trilha segue os mesmos padrões dos outros browsers descritos.

Para a implementação do mecanismo de navegação por trilhas foi necessária a implementação da classe trilha do Modelo de Contextos Aninhados e de uma classe chamada de *Perspective*, que relaciona o nó componente da trilha com os nós da sua perspectiva corrente. Na verdade, a classe trilha é composta por uma lista ordenada de objetos da classe *Perspective*. Para maiores detalhes sobre a implementação das classes do Modelo de Contextos Aninhados, deve-se consultar o Anexo 2.

O browser de trilha é um método das trilhas do sistema HyperProp e permite que se visualize e manipule a estrutura de uma trilha e se navegue por ela.