

# Capítulo 1

## Introdução

Uma das principais vantagens de sistemas hipertexto/hipermídia, ou de documentos não lineares, é sua habilidade de organizar seus documentos de diferentes maneiras, dependendo do ponto-de-vista (contexto) adotado. Entretanto, a promessa dos sistemas hipermídia de produzir informações multimídia complexas, ricamente interconectadas e cheias de referências cruzadas, na maioria das vezes implica na construção de documentos muito extensos e intratáveis.

Documentos hipermídia extensos, devido à sua complexidade, provocam certa dificuldade no que se refere à orientação do leitor, em relação ao conhecimento do tamanho total do documento e das conexões existentes entre os nós. Constitui-se, portanto, uma séria preocupação dos sistemas hipermídia em tentar minimizar esses problemas.

Apesar dos sistemas hipermídia terem amadurecido em termos de poder e sofisticação, muitos problemas ainda não estão resolvidos. Uma área que vem recebendo bastante atenção é a de técnicas de navegação eficazes para exploração de hiperdocumentos extensos e desconhecidos. Muitas técnicas foram desenvolvidas assumindo que a maioria dos usuários navega querendo localizar algum item específico de informação. Infelizmente, essas técnicas não se aplicam aos usuários que desejam apenas entender a estrutura global de um hiperdocumento extenso ou saber em que posição da estrutura eles se encontram durante o processo de navegação.

Existem várias formas de navegação pelos hipertextos, entre elas:

- percorrendo elos e passando de um nó para outro via a relação estabelecida;
- procurando nós por consulta, definidas através de uma linguagem apropriada;
- navegando em profundidade pelos nós, nos modelos que permitem composições;
- navegando por uma representação gráfica do hipertexto, tal como um mapa ou árvore;
- navegando por trilhas.

Parâmetros como grande número de nós, grande número de elos, muitas mudanças no hipertexto, tempo de resposta ruim para as ações do usuário, diferenças visuais insuficientes entre elos ou nós, e usuários não orientados visualmente se combinam para dificultar os mecanismos de navegação existentes.

Usuários desorientados precisam de informações de contexto para reestabelecerem a noção de localização. Em particular dois tipos de contexto são necessários: contexto espacial que responde à questão “para onde posso ir agora ?” e contexto temporal que responde à questão “como eu cheguei aqui ?”.

Para fornecer aos usuários a informação de contexto temporal, duas técnicas têm sido utilizadas: registrar o caminho percorrido e permitir que o usuário possa voltar atrás neste caminho, denominado trilha. Para voltar a ler o documento sem ser forçado a recriar exatamente os mesmos passos através dos elos percorridos anteriormente, pode-se navegar pela trilha criada através da história do último caminho percorrido por cada leitor.

Para resolver a questão do contexto espacial, a maneira mais direta é dar aos usuários um sentido de localização através de um mapa de todo o espaço de nós do documento e mostrar-lhes a sua posição, indicando de onde vieram e para onde podem ir. Entretanto, quando o número de nós e elos do hiperdocumento é grande, a apresentação do mapa, ao invés de melhorar a orientação do usuário, pode prejudicá-la ainda mais. Quando o diagrama apresenta muitas informações ao mesmo tempo, sua legibilidade e utilidade ficam comprometidas.

Outro problema fundamental da utilização de hipertextos é a dificuldade de se acostumar com a sobrecarga mental adicional para criar, nomear e gerenciar elos. Este problema também está presente no processo de leitura dos documentos, que oferece aos usuários um grande número de opções sobre quais elos seguir. Estas escolhas acarretam uma sobrecarga de decisões aos leitores, que pode ser eliminada quando o próprio autor define previamente os caminhos a serem seguidos.

Resumindo, os problemas de localização temporal e espacial em sistemas hipertextos são:

- desorientação: tendência a perder o sentido de localização e direção em um documento não linear;
- sobrecarga cognitiva: esforço adicional e concentração necessária para manter várias tarefas ou trilhas ao mesmo tempo [Conk87].

Estes problemas podem ser parcialmente resolvidos por melhoras no desempenho e no desenho da interface de sistemas hipertextos e por técnicas de filtragem das informações que serão exibidas aos usuários.

Neste trabalho, tenta-se minimizar o problema da desorientação do usuário explorando dois mecanismos de navegação: browsers e trilhas. O seu objetivo é propor a construção de ferramentas de auxílio à navegação em hiperdocumentos, melhorando a orientação do usuário e facilitando a edição da estrutura através de sua apresentação gráfica.

O browsers de documentos, ou simplesmente browsers, são ferramentas que exibem a estrutura de um hiperdocumento, ou parte dele, através de diagramas gráficos de nós e elos, permitindo que o usuário tenha informações sobre a estrutura global do hiperdocumento e orientação de quais possíveis caminhos pode seguir durante o processo de navegação. Entretanto, à medida que o número de nós e elos aumenta, os diagramas gráficos apresentados se tornam muito complexos, como por exemplo na Figura 1, podendo fazer com que a exibição do mapa global perca sua utilidade.

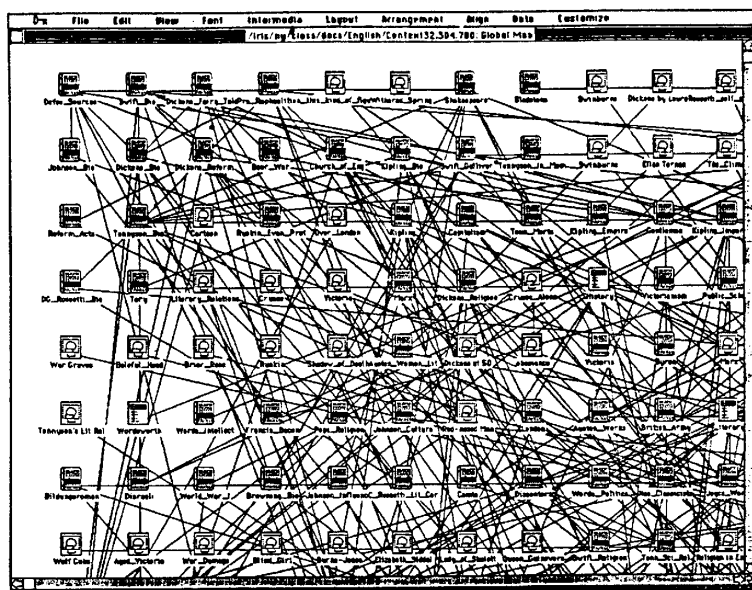


Figura 1: Mapa global de um hipertexto

Com o objetivo de melhorar a legibilidade dos mapas e garantir a eficácia dos browsers, foram desenvolvidas técnicas para filtragem das informações exibidas. Dentre elas, destaca-se o conhecido modelo de lentes de olho-de-peixe, proposto por Furnas [Furn86], que será utilizado como base inicial do presente trabalho.

As trilhas são caminhos pré-definidos sobre um hipertexto que permitem linearizar a ordem de leitura ou registrar os passos dos usuários durante um processo de navegação. Este tipo de recurso permite que os usuários voltem atrás no caminho percorrido ou percorram trilhas já definidas sobre um hipertexto no caso de ainda não estarem familiarizados com suas informações.

A definição de documentos hipermídia é baseada geralmente em dois conceitos familiares chamados nós e elos. Existem alguns modelos conceituais que permitem agrupar nós em nós de composição e também permitem composições aninhadas. Tais modelos hipermídia, ao permitirem que grupos de nós sejam organizados, hierarquicamente ou não, oferecem, entre outras facilidades, uma maneira melhor de ajudar a reduzir o problema da desorientação, como anteriormente mencionado.

Baseando-se em modelos conceituais hipermídia que oferecem composições aninhadas de nós, apresenta-se uma proposta de construção de browsers e trilhas que, à princípio, não está associada a nenhum modelo conceitual específico, sendo válida para qualquer sistema hipermídia que possua tal característica. Para ilustrar sua validade, foi feita uma implementação sobre o sistema hipermídia HyperProp, baseado no Modelo de Contextos Aninhados [SoCR95], em desenvolvimento pelo Grupo de Redes e Sistemas Multimídia do Departamento de Informática da PUC-Rio.

Inicia-se esta dissertação apresentando características de alguns dos principais sistemas hipermídia existentes no que diz respeito a soluções ou tentativas para solucionar o problema de desorientação do usuário, informalmente chamado “perdidos no hiper-espaço”.

Através da análise do que já foi feito nesta área de pesquisa, pretende-se identificar os recursos mais eficazes oferecidos por esses sistemas e incorporá-los nesta proposta. Este estudo é feito no Capítulo 2.

Logo em seguida, apresenta-se um estudo de algumas técnicas para construção de diagramas gráficos. Estas técnicas têm contribuição importante nesse trabalho, pois para construir diagramas gráficos da estrutura de nós e elos dos hiperdocumentos, necessita-se de estudos para apresentar estruturas de grafos automaticamente. Este tema será abordado no Capítulo 3.

Após estudar e analisar os recursos de auxílio à navegação oferecidos pelos principais sistemas hipermídia existentes e algumas técnicas para realizar o layout automático de grafos, apresenta-se finalmente a proposta de construção do mecanismo de navegação baseado em browsers e trilhas para modelos hipermídia com composições aninhadas de nós. São ressaltadas algumas características principais desses modelos, utilizando como base o Modelo de Contextos Aninhados utilizado pelo sistema hipermídia HyperProp, e são explicitadas as técnicas utilizadas para construir os browsers e trilhas sobre este sistema. No Capítulo 4, portanto, será abordado o foco principal deste trabalho.

Depois de apresentar a proposta para minimizar o problema da desorientação do usuário ao navegar pelo “hiper-espço”, são descritos alguns detalhes de implementação das ferramentas sobre o sistema HyperProp, explicitando os recursos oferecidos aos usuários, o que será feito no Capítulo 5.

Finalizando, é feita uma avaliação crítica dos resultados obtidos, ressaltando as contribuições deste trabalho, realizando uma comparação com trabalhos relacionados e indicando quais serão os principais tópicos a serem abordados como trabalhos futuros. Estas conclusões serão apresentadas no Capítulo 6.

## Capítulo 2

### Browsers e Trilhas

Browsing é uma atividade canônica para usuários hipermídia. A habilidade de navegar rapidamente no sistema é crítica para sua usabilidade, principalmente para sistemas muito grandes.

O browser é um componente importante dos sistemas hipermídia, pois à medida que um hipertexto se torna mais complexo, é muito fácil um usuário se perder ou ficar desorientado. É função do browser mostrar todo o documento ou parte dele como um grafo, fornecendo uma importante medida de contexto e espaço. Isto ajuda ao usuário a saber quais nós ele está visualizando e como eles estão relacionados com seus vizinhos no grafo. Usar o browser pode ser comparado a usar o tato e a visão quando procuramos uma certa página em um livro. Às vezes lembramos como a página era, mas não lembramos qual o seu número ou as palavras-chave que possam identificá-la através do índice. O browser pode ser consultado desta forma, quando o leitor esqueceu de tudo, a não ser da aparência ou localização do nó [Conk87].

Os browsers se apóiam no processamento visual/espacial altamente desenvolvido do sistema visual humano. Localizando nós e elos em um espaço de duas ou três dimensões, utilizando propriedades para diferenciação visual (cor, tamanho, forma, textura) e mantendo algumas similaridades com nosso ambiente físico (dois objetos não ocupam o mesmo lugar no espaço), os projetistas de browsers criam ambientes razoáveis para visualização. Entretanto, não existe uma topologia natural para um espaço de informações. Desta forma, até que o usuário se torne familiarizado com um dado hipertexto, ele está por definição desorientado.

Uma tentativa de melhorar a orientação do usuário é utilizar o layout espacial, ou melhor, fixar uma posição para cada nó permitindo que o usuário desenvolva uma noção de localização na estrutura global. Este tipo de layout induz a comparação, por exemplo, assumindo que nós mais relacionados estão posicionados juntos e nós sem relação estão bem separados.

No caso de redes muito extensas, o tamanho do mapa se torna um problema. Para sistemas com dados organizados hierarquicamente isto pode ser minimizado agrupando ou expandindo partes da hierarquia como desejado.

A construção de browsers gráficos da estrutura de documentos como ferramentas de auxílio à navegação e que também possam oferecer funções de edição da estrutura dos hiperdocumentos não é algo trivial para projetistas de sistemas hipermídia. Daí, o interesse em estudar os trabalhos nesta área e analisar quais seriam as melhores técnicas para a

implementação de tais ferramentas em modelos conceituais hipermídia que permitem composições aninhadas de nós.

Além do browser, outra ferramenta importante para hipertextos, e também alvo de bastante interesse, é a definição de trilhas [Zell89], que são travessias ordenadas de alguns elos do documento. Através delas, os autores podem fornecer ordens de leitura que auxiliam leitores não familiarizados com o material informativo a navegar pelo hipertexto, ou determinar uma ordem apropriada de apresentação para uma certa audiência. Os usuários sentem-se menos desorientados quando seguem uma trilha já definida, pois têm limitado o número de opções para percorrer o documento.

As trilhas fornecem informações importantes sobre o contexto temporal, auxiliando o usuário a responder a questão “como eu cheguei aqui?”, logo representam recursos fundamentais para melhorar o sentido de localização durante o processo de navegação.

Os sistemas devem fornecer recursos para facilitar a criação, seleção de trilhas, assim como sua navegação. Um mecanismo de trilhas efetivo deve satisfazer três requisitos principais: aumentar o poder expressivo dos autores; ajudá-los a criar e modificar trilhas e ajudar os leitores a encontrá-las e navegar por elas de maneira flexível. Ainda mais, deve prover armazenamento e execução eficientes das estruturas.

Um mecanismo de trilhas deve incluir um editor que permita que o autor crie e edite trilhas de maneira rápida e fácil, rearranjando, acrescentando ou eliminando entradas. Estas operações de edição tomam formas diferentes dependendo da representação da trilha. Estas podem ser representadas por seqüências, grafos dirigidos ou “programas” (texto linear). Seqüências e programas não têm elos explícitos.

As trilhas devem se manter consistentes quando o autor edita o material informativo. Por exemplo, se o autor elimina algum nó do hipertexto, as entradas correspondentes em todas as trilhas devem ser removidas. Ou, por outro lado, se a versão corrente de um nó faz parte de uma trilha e essa versão é atualizada, a trilha também deverá referenciar a nova versão.

Os leitores devem ser capazes de encontrar trilhas relevantes de maneira simples, ou seja, o nó corrente deve mostrar se uma trilha começa por ele ou o atravessa dentro do contexto em questão. Além disso, devem existir formas gerais para localizar trilhas relevantes.

Existem vários métodos para exibição de trilhas:

- controle passo-a-passo, onde o leitor começa uma trilha e fornece o comando “Próximo” repetidamente para segui-la;
- controle automático, permite que o leitor comece a trilha e continue automaticamente a percorrer suas entradas. Um sistema ideal deve permitir que o leitor ajuste a velocidade de exibição;
- controle por browsing, onde o leitor pode percorrer as entradas de maneira arbitrária, selecionando-as individualmente.

A seguir, serão descritas características dos principais sistemas hipermídia existentes em relação a mecanismos de navegação baseados em browsers gráficos e trilhas. São eles o Neptune, o Electronic Document System, o Scripted Documents, o NoteCards, o Thoth-II, o Hypercard, o KMS, o Intermedia, o GRAB, o Interactive Graphical Document, o CYBERMAP, o Storyspace e o SHADOCS.

## 2.1 Neptune

O Neptune [CaGo88] [DeSc86] é um sistema hipertexto desenvolvido em uma arquitetura em camadas. A principal camada é um servidor baseado em transações chamado HAM - Hypertext Abstract Machine - [CaGo88]. O HAM constitui o modelo genérico de hipertexto (vide Figura 2) que provê mecanismos de armazenamento e acesso para nós e elos. As camadas adicionais são construídas em cima do HAM.

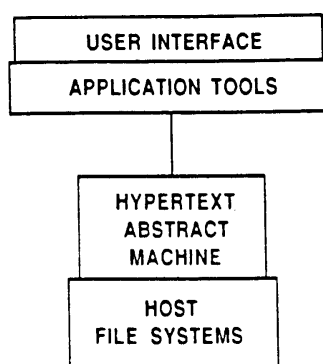


Figura 2: Modelo Genérico do HAM

A camada de interface com o usuário provê browsers para edição e visualização de hipertextos e do conteúdo dos nós. Existem três tipos primários de browsers:

- **Browser Gráfico:** provê uma visão ponto-a-ponto do subgrafo de nós e elos, permitindo que o usuário os selecione. Cada nó é representado por um ícone que consiste de um nome dentro de um retângulo. As duas regiões de edição de texto na parte inferior do browser são usadas para a definição dos predicados de visibilidade dos nós e elos (vide Figura 3).

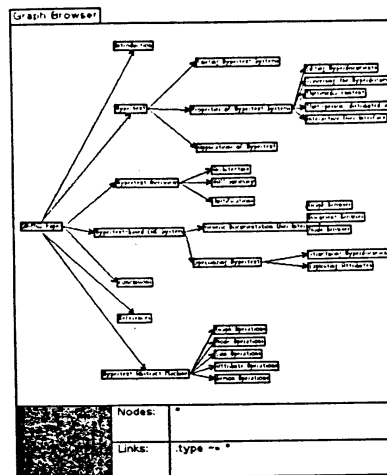


Figura 3: Neptune - Browser Gráfico

- Browser de Documento: projetado para simplificar a manipulação de documentos estruturados hierarquicamente. A parte superior fornece uma lista nominal da estrutura hierárquica dos nós e a parte inferior é um browser de nó (exibidor do nó) usado para visualizar o conteúdo de um dos nós listados, conforme visto a seguir:

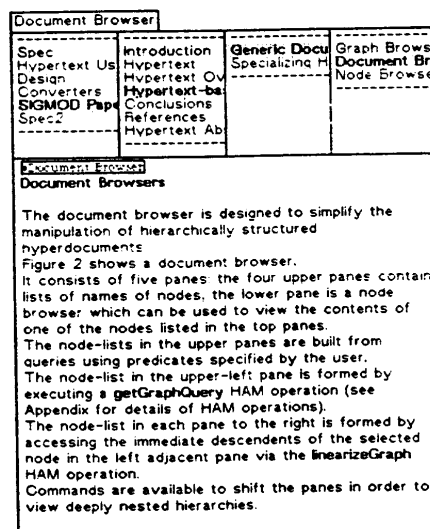


Figura 4: Neptune - Browser de Documento

- Browser de Nó: exibe um nó individual em um hipertexto, possibilitando a edição de seu conteúdo e dando suporte à criação de elos (vide Figura 5).



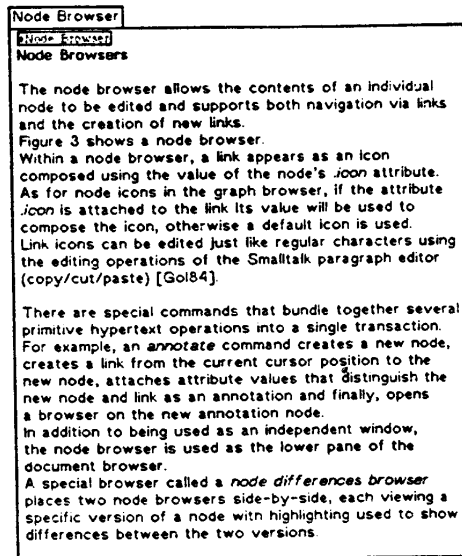


Figura 5: Neptune - Browser de Nó

Há outros tipos de browsers no Neptune, são eles: browser de atributos, de versões, de diferenças entre nós e de demonstrações (demo browser). O browser de diferenças entre nós posiciona dois browsers de nós lado a lado, cada um exibindo um versão específica do nó, e realça todas as diferenças entre as duas versões.

Além do browser gráfico, o Neptune mantém uma trilha do que foi feito pelo usuário. Esta trilha informa o caminho que foi seguido e permite que este seja percorrido por outros usuários [DeSc86].

## 2.2 Electronic Document System

O Electronic Document System [YaMe85], terminado em 1982, foi o primeiro sistema hipermídia da Brown University. Os seus aspectos mais interessantes incluem a representação gráfica da rede de informações que está disponível no modo autoria e duas formas de mapas oferecidas no modo de leitura.

O primeiro tipo de mapa oferecido aos leitores mostra uma história do caminho percorrido no eixo do tempo que miniaturiza as páginas visitadas e permite que se selecione qualquer ícone para retornar à página já acessada.

O segundo tipo de mapa ajuda a orientar o leitor espacialmente. No centro da tela, o sistema mostra a miniatura da página corrente visualizada pelo leitor. Do lado esquerdo, apresenta miniaturas de todas as possíveis páginas de onde o leitor pode ter vindo e do lado direito, todas as páginas para as quais ele poder ir. Novamente, qualquer ícone pode ser selecionado para acessar a página desejada.

## 2.3 Scripted Documents

O sistema Scripted Documents [Zell89] foi desenvolvido na Xerox PARC e roda em estações de trabalho pessoais Dorado em ambiente distribuído conectado pela Ethernet. Ele é um sistema hipertexto diferente, pois seu único mecanismo de ligação é através de trilhas.

Existem três tipos básicos de trilhas:

- Uma trilha seqüencial é uma seqüência ordenada de entradas que pode ser usada para apresentar uma progressão ordenada ou simplesmente colecionar um conjunto desordenado de entradas. A maioria das implementações existentes permite que uma entrada apareça repetidamente em uma mesma trilha.
- Uma trilha branching contém caminhos que o leitor escolhe manualmente, agindo como uma sub-rede do hipertexto e fornecendo opções para customização da trilha de acordo com interesses do leitor.
- Uma trilha condicional contém caminhos nos quais o sistema avalia testes fornecidos pelo autor para determinar que direção seguir. Estendendo esta idéia para um paradigma de programação mais completo, outros tipos de trilha aparecem:
  - trilhas procedurais, que permitem que uma entrada seja uma outra trilha, suportando modularização e reutilização de trilhas;
  - trilhas programáveis, que são trilhas condicionais que podem armazenar valores em variáveis para formar loops ou guardar informação para uso futuro;
  - trilhas variáveis, que contêm elos variáveis nos quais a próxima entrada é localizada dinamicamente, talvez sendo computada por uma entrada anterior ou envolvendo pesquisa pelo hipertexto;
  - trilhas paralelas, que são trilhas que podem ser executadas simultaneamente.

Em sua forma mais simples, uma entrada se refere a um nó no hipertexto, e uma trilha fornece uma maneira de apresentar uma seqüência de nós. Uma trilha também pode ser constituída por seqüência de elos, porém isto impõe um requisito: o destino de uma entrada “i” deve ser igual a origem de uma entrada “i+1”.

O sistema fornece trilhas condicionais e programáveis, controle automático de exibição, suporte para navegação para autores e leitores, entradas ativas com amplo escopo de ações e mídias diversas, incluindo voz.

As trilhas possuem entradas ativas, chamadas scripts, que são localizações de documentos associadas a ações e informações temporais. Um script contém três partes:

- um conjunto de documentos;
- um conjunto de entradas que associam localizações ou objetos dentro dos documentos a ações e informações temporais;
- uma especificação de trilha, conhecida como script header, que é um programa cuja complexidade depende das necessidades do autor.

O sistema alerta os autores quando uma modificação em um documento afeta a integridade de um script, e ajuda os leitores a encontrar scripts de possível interesse, sinalizando-os nos documentos correspondentes ou exibindo uma lista com todos os scripts que começam no documento ou o atravessam. O usuário também pode acessar um script pelo nome ou por consultas.

O Scripted Documents fornece um mecanismo de exibição de trilhas bastante rico, através de controle automático, passo-a-passo ou por browsing.

## **2.4 NoteCards**

NoteCards [Hala88] foi desenvolvido por uma equipe da Xerox PARC com o objetivo de transformar uma coleção caótica de pensamentos não relacionados em uma interpretação de idéias ordenada e integrada. No sistema são fornecidos mecanismos para construção de redes de cartões eletrônicos (nós NoteCard) conectados por elos tipados.

Existem dois tipos especiais de cartões que auxiliam o usuário na gerência da rede de informações, os fileboxes e os browsers (vide Figura 6).

Os fileboxes são cartões utilizados para organizar e criar categorias em grandes coleções de cartões interligados. Um filebox contém outros cartões ou outros fileboxes, representando uma estrutura organizacional hierárquica da rede. O sistema exige que cada cartão esteja contido em pelo menos um filebox.

O browser é um cartão que contém um diagrama estrutural da rede de cartões (vide Figura 7). Este diagrama pode ser visualizado em vários níveis de detalhe, pois a informação pode ser filtrada de acordo com os tipos de nós e elos. Os tipos são indicados pela aparência dos nós e elos na tela [HaMT87].

Já que o NoteCards tem capacidade para múltiplas janelas, o usuário pode visualizar o browser continuamente enquanto explora o hipertexto e lê os cartões. O browser também pode ser usado para navegação, pois atravessando um elo, o usuário obtém o cartão correspondente. Além disso, tem a finalidade de facilitar a edição do hipertexto, permitindo modificar nós e elos e experimentar a alteração antes de realizá-la efetivamente.

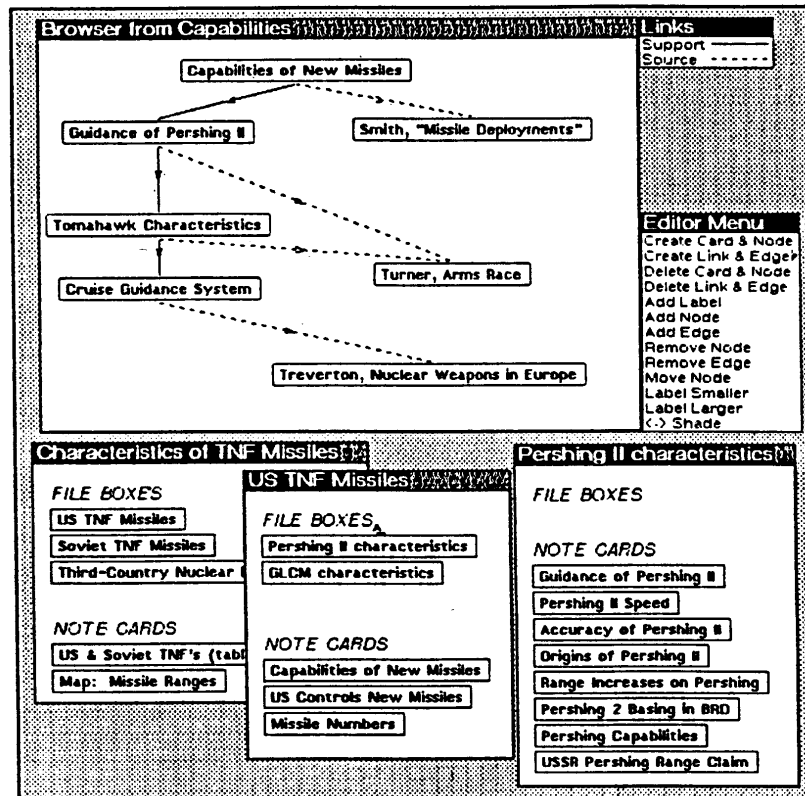


Figura 6: NoteCards - Exemplos de Browsers e Fileboxes

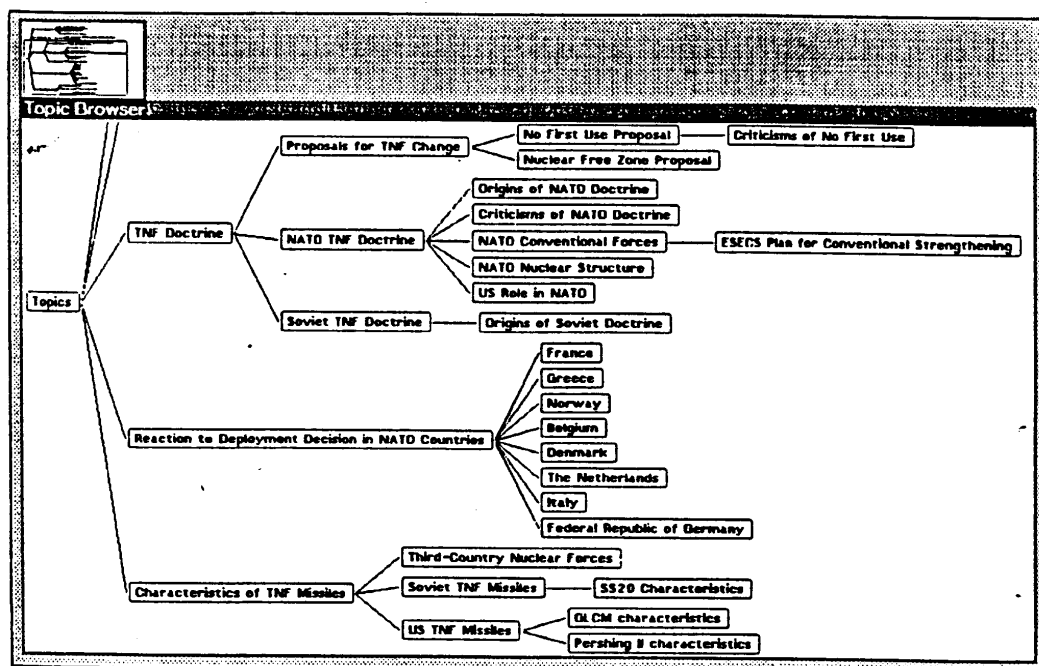


Figura 7: NoteCards - Browser de um Filebox

Mais tarde foi desenvolvido um conjunto de extensões para o NoteCards com o objetivo de melhorar o problema da desorientação.

Na chamada History List (vide Figura 8), é mantida uma lista ordenada com cada nó acessado pelo usuário. Esta lista pode ser vista enquanto o leitor navega pelos elos e nós. Ele pode também fazer anotações em qualquer lugar da janela da History List. Através desse recurso, o usuário pode selecionar um item e obter um minibrowser correspondente. Se um nó foi visitado na sessão corrente, este nó é acompanhado do sinal de mais (+) no minibrowser (um sinal para cada vez que o nó for visitado).

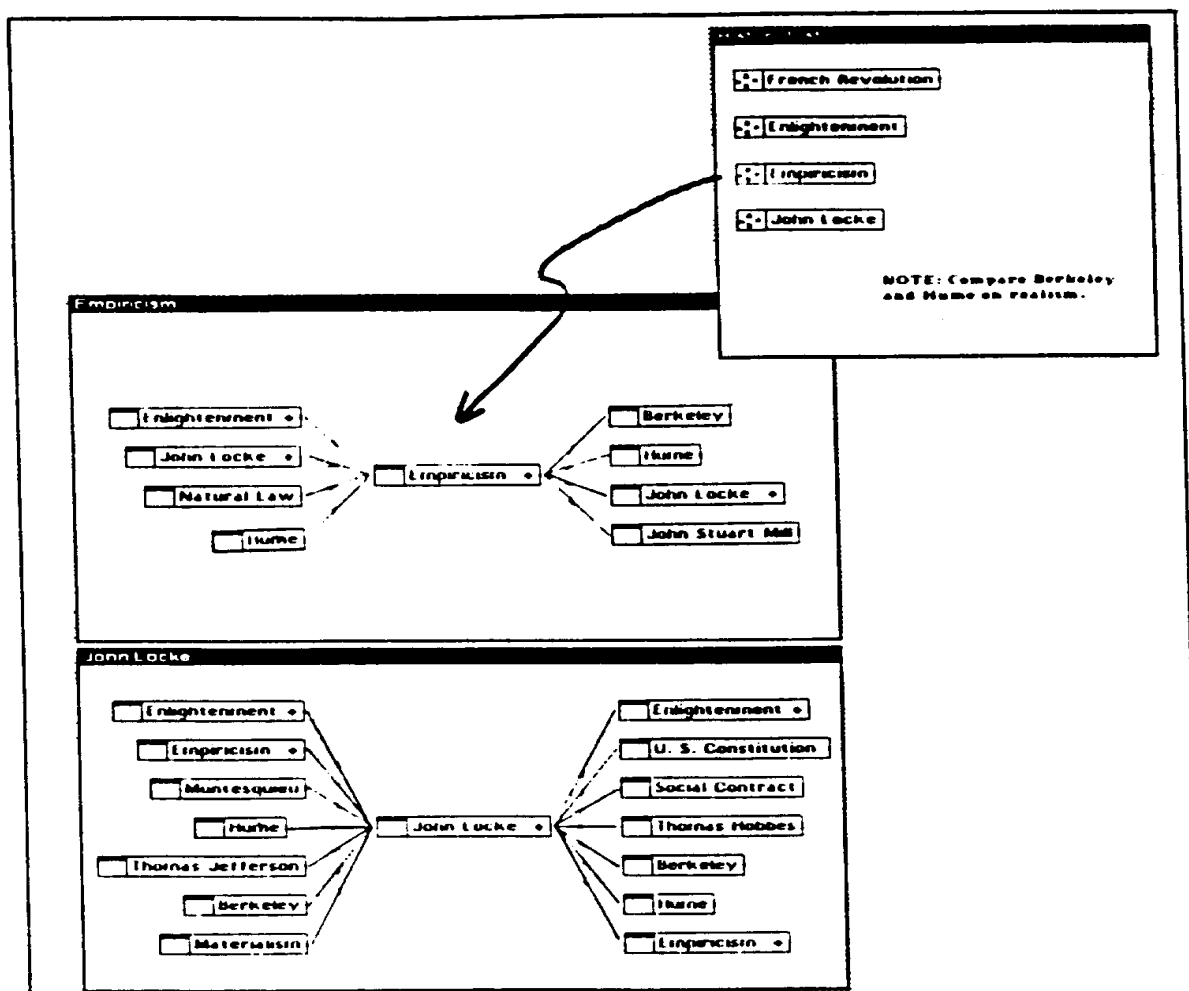


Figura 8: NoteCards - History List e minibrowsers correspondentes

Um outro recurso adicionado foi a History Tree (vide Figura 9). Ao contrário da History List, que é linear, esta outra extensão é hierárquica. O seu propósito é tentar mostrar ao usuário como ele percorreu um conjunto de nós ligados e quais nós foram examinados várias vezes (sinalizado pelo símbolo “+”). Assim como a History List uma versão da History Tree pode ser salva e comentada com textos e gráficos [UtYa89].

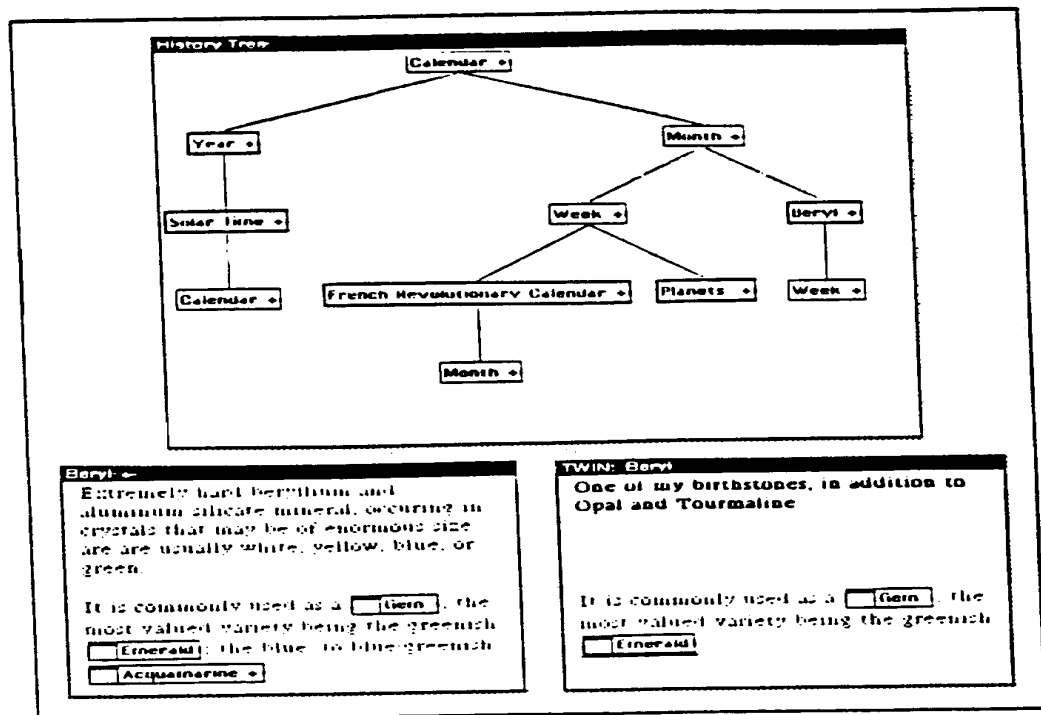


Figura 9: NoteCards - History Tree

Além da History List e da History Tree, existe um mecanismo de comunicação entre o autor e os futuros leitores de uma rede de cartões, chamado Guided Tours [MaIr89]. Esta facilidade provê um método interativo, baseado em grafos, para representar a estrutura de uma rede de cartões, onde o autor constrói uma “viagem” pela rede e o leitor pode controlá-la de várias formas, navegando passo-a-passo ou realizando browsing pelo grafo. Porém, não é fornecido nenhum mecanismo para controle automático da navegação.

## 2.5 Thoth-II

Thoth-II [UtYa89] foi desenvolvido na University of North Carolina e apresenta uma proposta diferente dos outros sistemas. Ao invés de fornecer um mapa global da rede, fornece um que é atualizado dinamicamente à medida que o usuário percorre os nós. Quando o usuário seleciona um nó, o diagrama se expande para mostrar os elos conectados a ele (vide Figura 10). A expansão cria novas instâncias de todos os nós conectados, sem reutilizar os que já estavam desenhados. Conseqüentemente, não há cruzamento de elos. Entretanto, à medida que o usuário navega, o mapa fica cada vez maior, o que não se torna interessante para redes grandes.

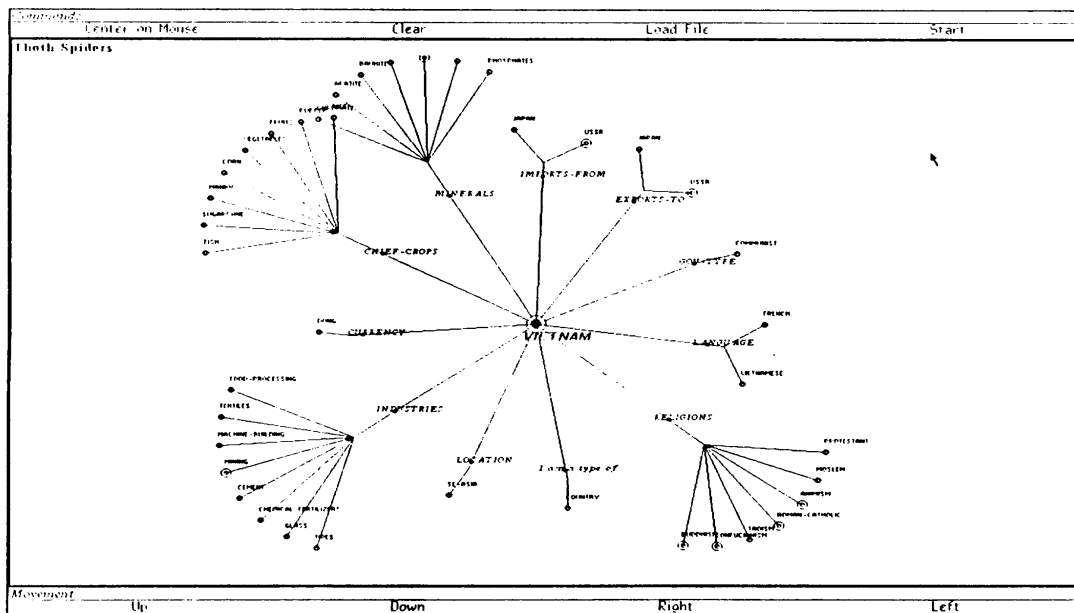


Figura 10: Diagrama do Thoth II

## 2.6 Hypercard

O Hypercard [Will87] foi desenvolvido pela Apple Computer Inc. para a plataforma Macintosh. O que ele oferece de interessante é um artifício chamado Recent, que mostra miniaturas de todos os nós já acessados pelo usuário (vide Figura 11). A idéia é que o leitor pode não se lembrar do nome do nó a que deseja retornar, mas sim de sua aparência. O único detalhe é que o Recent não exibe o mesmo nó mais de uma vez, mesmo que ele tenha sido visitado mais vezes. Isto faz com que este recurso não reflita exatamente a história do caminho percorrido pelo usuário [UtYa89][Niel90].

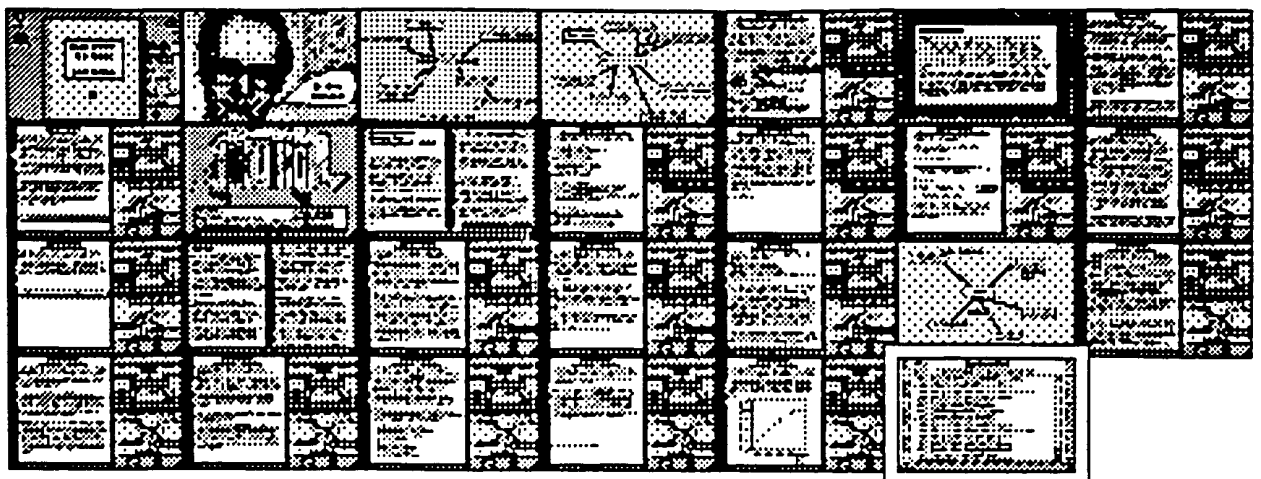


Figura 11: Hypercard - Recent

Uma outra ferramenta implementada em cima do Hypercard, descrita em [LaMa91], é uma técnica para ajudar os usuários a se orientarem através de uma navegação rápida pelo hipertexto, chamada de vôo.

Esta ferramenta é análoga a virar páginas de um livro com uma notável diferença: as páginas não precisam estar em ordem linear. O objetivo não é digerir o conteúdo das páginas, mas ganhar um noção de organização, tamanho, profundidade, nível, detalhes, etc. O usuário deve ser capaz de voar em diferentes ordens e controlar o vôo interativamente.

O vôo também pode ser usado para navegar rapidamente de um lugar para outro. A operação não pretende substituir outras ferramentas de navegação, mas sim ser uma nova ferramenta.

Uma parte essencial de qualquer ordem de travessia é a habilidade de voltar atrás e parar o vôo a qualquer momento. Para isto, deve-se manter uma trilha de todos os nós visitados e ser capaz de voltar atrás. Não somente a trilha de nós, mas todas as ações feitas pelo usuário devem ser salvas, e o resultado do comando “Desfaz” (Undo) é voltar exatamente ao estado em que o usuário se encontrava antes de executar a última ação.

A existência de recursos como som ou animação pode produzir efeitos colaterais que consomem tempo ou distraem o usuário da travessia sendo feita. Por isso, deve existir uma maneira de desabilitar estes recursos enquanto o leitor voa pelo hipertexto. Quando o vôo estiver terminado, os recursos voltam a estar disponíveis.

O vôo do Hypercard fornece recursos para o usuário escolher a velocidade e a ordem de travessia do grafo. Existem cinco ordens de travessia disponíveis: por amplitude, por profundidade, seqüencial, seqüencial completa e externa. As buscas por amplitude e profundidade são as padrões para grafos. A seqüencial é na ordem em que os cartões estão armazenados. A seqüencial completa atravessa todos os cartões enquanto a seqüencial só passa pelos não acessados. A externa implementa a parada do vôo, passando o controle de volta ao mecanismo de navegação do Hypercard. Como o vôo não está integrado ao Hypercard, não é possível armazenar o caminho percorrido durante uma travessia externa.

## **2.7 KMS - Knowledge Management System**

O KMS [AkMY88] é um sistema hipermídia distribuído para estações de trabalho desenvolvido na Carnegie-Mellon University. Os nós do KMS são chamados *frames*, onde o usuário pode armazenar texto, gráfico e imagens. Elos são tratados como componentes de nós, como nos sistemas NoteCards e Hypercard, e diferentemente dos que representam nós e elos separadamente, como o Intermedia, por exemplo.

Não existem limitações entre editar e navegar. O usuário pode manipular diretamente o conteúdo de um frame a qualquer momento. Quando um usuário sai de um frame que acabou de editar/criar, as mudanças são salvas automaticamente.



Existem duas maneiras geralmente utilizadas por sistemas hipermídia para apresentar os nós na tela: cada nó em janela separada, com várias janelas se sobrepondo, talvez até de diferentes tamanhos; ou apenas uma janela, onde cada nó é expandido de uma vez. A opção do KMS é diferente: dois nós, cada um ocupando metade da tela ou apenas um ocupando a tela toda, dependendo da escolha do usuário. Não existem outras possibilidades. Quando um usuário seleciona um item ligado a outro frame, o frame corrente é substituído pelo novo. Devido à capacidade do KMS de percorrer elos rapidamente, acredita-se na utilização da dimensão de tempo para manter nós relacionados juntos (time-multiplexing) ao invés de deixá-los visíveis na tela ao mesmo tempo (space-multiplexing).

O sistema provê navegação rápida como uma alternativa para o scroll que não é fornecido. Existem comandos para voltar atrás, o que é uma atividade bastante freqüente.

O KMS não fornece um browser gráfico, mas apenas uma visão linear da hierarquia de frames. Acredita-se que tais browsers têm valor limitado, exceto para estruturas grandes e essencialmente não hierárquicas. Visões gráficas adicionais podem ser úteis, mas não se sabe se compensa a complexidade acrescentada.

Para auxílio à navegação, existe ainda um mecanismo de procura de texto em qualquer hierarquia de frames.

## 2.8 Intermedia

O Intermedia [UtYa89] [Meyr86] foi desenvolvido no Institute of Research in Information and Scholarship na Brown University com o intuito de ajudar autores e leitores a manusear a complexidade de uma rede de idéias conectadas.

O sistema define blocos conectados por elos formando uma rede de conexões inter e intra-documentos. Estas conexões são armazenadas nas chamadas webs, onde todos os elos são bidirecionais. Os documentos são os nós e os blocos as âncoras do sistema.

O Intermedia fornece um recurso chamado Web View (vide Figura 12) que é uma ferramenta para visualização e navegação nas webs com um mínimo de confusão e desorientação do usuário. Os fatores-chave no sucesso da Web View são [UtYa89]:

- a combinação entre o registro do caminho percorrido pelo usuário e uma mapa dos elos disponíveis no momento;
- uma linha de escopo que indica o número de documentos e elos da rede.

Além disso, ela oferece três importantes recursos:

- a atualização dinâmica do mapa do usuário;
- a previsão de elos, para dar informação sobre o destino de um elo sem precisar segui-lo;
- a permissão para o usuário abrir documentos e seguir elos através da Web View.

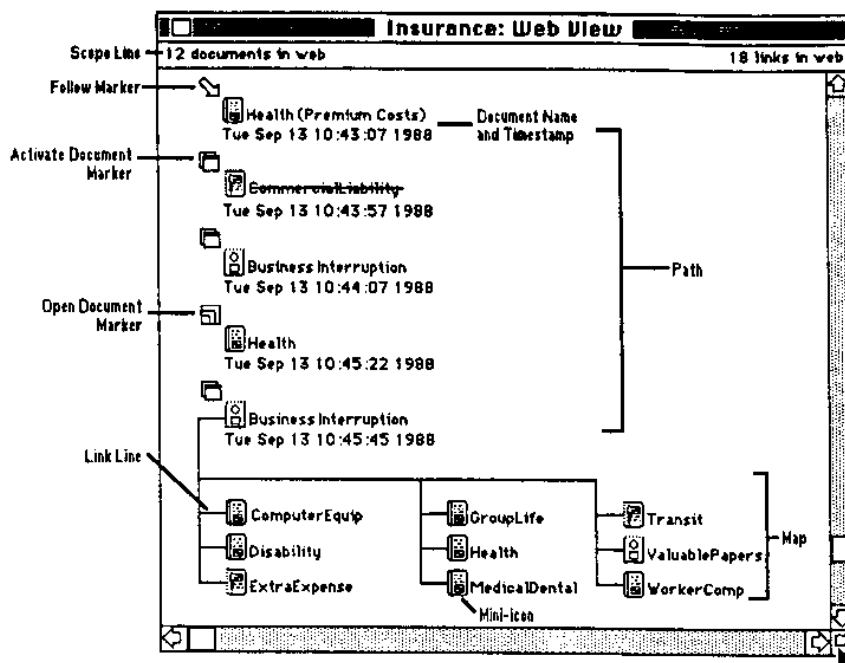


Figura 12: Intermedia - Web View

A atualização dinâmica da Web View é um recurso importante, pois assegura que a informação esteja sempre disponível.

O mapa provê a informação de contexto espacial, respondendo ao usuário a questão “para onde posso ir agora ?” e a linha de escopo, apesar de fazê-lo de forma rudimentar dá uma idéia do tamanho da web, indicando o seu número total de documentos e elos.

O algoritmo de layout do mapa considera o tamanho da janela, para que seja mostrado o maior número de documentos na tela, nas posições vertical (vide Figura 13) ou horizontal (vide Figura 14).

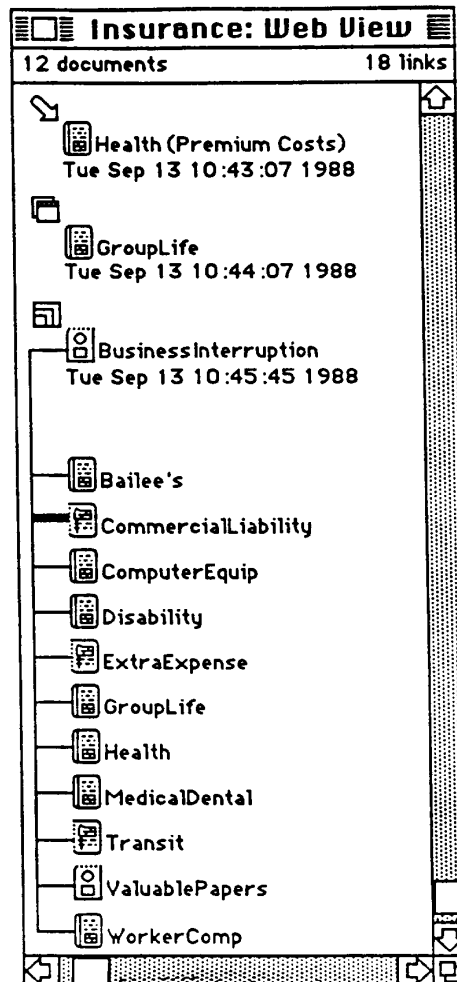


Figura 13: Web View - Layout Vertical

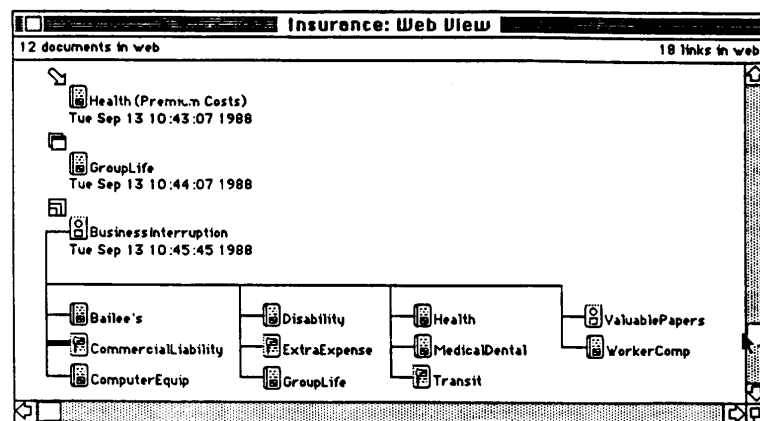


Figura 14: Web View - Layout Horizontal

Cada usuário tem um caminho único para cada web no sistema, sendo uma simples lista linear da atividade do usuário. O caminho permite navegar por ações passadas e reabrir

um documento já acessado. Ele é salvo toda vez que a web for fechada e restaurado da próxima vez que ela for aberta pelo usuário.

O caminho provê a informação de contexto temporal, mostrando ao usuário como o estado atual do sistema foi atingido e fornecendo um mecanismo conveniente para recuperar documentos já acessados.

Para abrir um documento através da Web View, basta selecioná-lo e acionar o comando “Abrir” do menu ou clicar duas vezes no ícone do documento. Além disso, selecionando um elo e acionando o comando “Seguir” do menu ou clicando duas vezes sobre a linha, o usuário percorre elos do documento corrente para outro documento relacionado.

Antes de definir a proposta final da Web View, a equipe do Intermedia desenvolveu diversas maneiras de implementação que ajudaram a orientar a decisão tomada. Estas outras implementações serviram como experiência e valem a pena ser ressaltadas.

O Intermedia foi inicialmente projetado com três tipos de Web View: um mapa global, uma mapa local e uma mapa de trilha. O mapa global (vide Figura 15) mostrava todos os documentos da web e os elos entre eles, sem entrar no detalhe dos blocos que compõem os documentos. Por essa causa, apenas uma linha é desenhada entre um par de documentos, não importando quantos elos os documentos compartilham. Apesar do mapa global ser útil no caso de webs pequenas, para estruturas grandes e complexas os mapa locais tinham maior utilidade. Estes mapas focalizavam um documento especificado pelo usuário e os documentos aos quais ele estava ligado (vide Figura 16). O mapa de trilha era exatamente igual ao mapa local, só que atualizava dinamicamente o documento em foco à medida que o usuário os acessava.

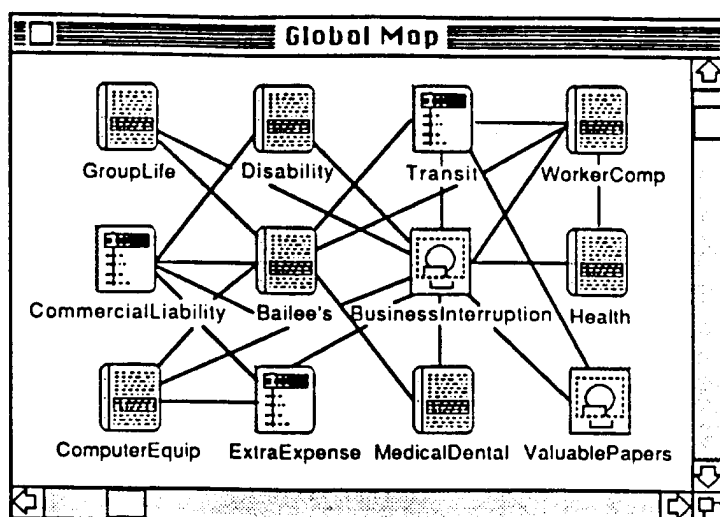


Figura 15: Intermedia - Mapa Global

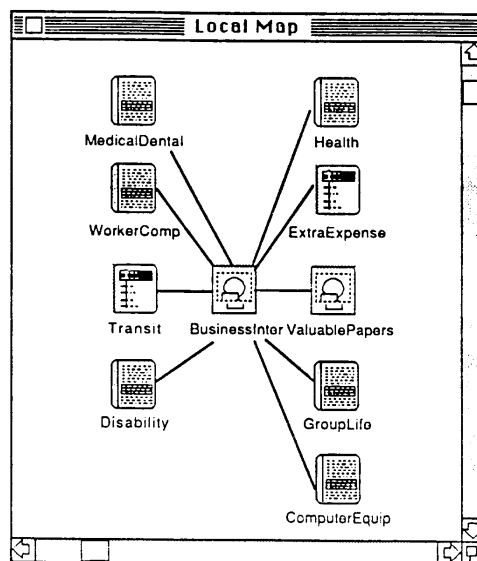


Figura 16: Intermedia - Mapa Local

Mais tarde foi tomada a decisão de implementar apenas uma única visão da web ao invés de fornecer três opções, só o mapa de trilha era oferecido.

Baseando-se na questão de contexto espacial, voltou-se à idéia de ter um mapa global para cada web, dando para cada documento uma posição fixa em um espaço bidimensional. Entretanto, o tamanho da janela para webs muito grandes tornou-se um problema, o que deixou novamente de lado esta idéia.

Outra estratégia foi baseada na visão olho-de-peixe. Seria mostrado na tela o quanto coubesse do mapa e esta janela deveria ser movida dinamicamente para que contivesse sempre o documento corrente (documento em foco). Apesar desta estratégia resolver alguns problemas do tamanho do layout, restaram dúvidas de até que ponto ela oferece uma imagem cognitiva do mapa. Por outro lado, essa técnica falhou ao resolver o problema do layout espacial, pois o sistema não poderia decidir como posicionar os documentos sem requerer uma participação efetiva do usuário.

Outra forma foi deixar que o usuário determinasse a posição do documento. Porém, isto não deu certo, pois quem garante que vários usuários compartilhando a mesma web concordam com a localização de cada documento definida por um usuário qualquer.

A estratégia mais óbvia foi definir a distância entre dois documentos como o menor número de elos que deve-se seguir para ir de um documento a outro. O que é chamado de estratégia para posicionamento por mínimo de elos (*minimal links placement strategy*). Obviamente a precisão da posição teria que ser sacrificada para prevenir a sobreposição de ícones no mapa. Entretanto também foram encontrados alguns obstáculos. Primeiro, à medida que novos elos e nós são criados, o mapa deve ser reorganizado, e realizar estas atualizações em tempo real é desafiante. Segundo, a mudança de posição dos documentos quando o mapa é atualizado, tira as vantagens esperadas de um layout espacial, não ajudando a memória do usuário.

Estes problemas fizeram com que uma estratégia semi-espacial fosse considerada, na qual só os documentos considerados chaves pelo usuário teriam uma posição fixa. Os outros flutuariam no espaço e apareceriam perto de qualquer documento quando este fosse o documento em foco. Uma dificuldade encontrada foi que a opinião dos usuários pode variar, não concordando com os mesmos documentos-chave.

No final, chegou-se à conclusão de que um mapa global da rede hipermídia, pelo menos no caso do Intermedia, não era factível. Decidiu-se focalizar a atenção no contexto local, fornecendo ao usuário ferramentas para navegação e não comparação, dando origem a proposta final da Web View, que foi descrita no início da seção.

## 2.9 GRAB

O GRAB [RDMM+87] é um browser gráfico desenvolvido na University of California, Berkeley, que pode ser utilizado para navegar por grafos dirigidos ou pode ser integrado ao mecanismo de navegação de uma aplicação específica.

O sistema exibe os nós do grafo por meio de ícones e os arcos através de linhas entre os ícones origem e destino. Tipos de nós diferentes podem ser exibidos com ícones diferentes e nomes podem ser dados aos arcos.

O usuário examina um grafo realizando scroll horizontal e vertical ou zoom em determinada parte do grafo. O GRAB também provê operações de edição que permitem que o usuário insira ou elimine nós e arcos ou mude os respectivos nomes. Além dessas operações de navegação e edição, o usuário pode invocar um exibidor específico para um nó, como por exemplo, um editor de texto.

O sistema determina o layout do grafo automaticamente, minimizando o número de arcos que se cruzam e posicionando os nós conectados juntos. O algoritmo implementado é baseado no que foi desenvolvido por Sugiyama [SuTT81] e será descrito na Seção 3.1.

Para facilitar a visualização de um determinado nó e seus referidos arcos, foi implementada a possibilidade do usuário ressaltar os arcos que chegam em um vértice ou saem dele, facilitando o entendimento do diagrama no caso de grafos muito grandes.

Com a implementação do sistema GRAB, os autores verificaram que o esquema de layout não era adequado para aplicações interativas, pois não possuía desempenho adequado. Com o objetivo de melhorar o tempo de resposta para desenhar os grafos e a qualidade de layout para estruturas grandes, os autores desenvolveram o sistema Compoze [MeRH91] que utiliza um novo algoritmo com a técnica “dividir-para-conquistar” para realizar o layout. Subdividindo o grafo grande em subgrafos, realizando os layouts em separado e juntando-os para criar o layout do grafo completo, o sistema Compoze produz layouts de melhor qualidade em menos 20% de tempo que o algoritmo de [SuTT81].

## 2.10 IGD - Interactive Graphical Document

O IGD [Fein88], Interactive Graphical Document, foi um ambiente de hipertexto experimental que suportava criação e exibição de estruturas de grafos arbitrárias através de estruturas hierárquicas de árvores.

Os nós no IGD são chamados páginas e representam as folhas da árvore. Os nós internos são chamados capítulos e podem conter páginas e capítulos aninhados recursivamente. O nível mais alto do documento também é considerado um capítulo, então, cada capítulo pode ser considerado um subdocumento. A estrutura de grafo é obtida por elos que conectam páginas. Estas são, além das folhas da árvore, os nós do grafo dirigido.

Capítulos e páginas são representados através de retângulos no diagrama. Capítulos têm bordas pretas, enquanto páginas têm bordas brancas. Elos são representados por setas entre as páginas.

Para apresentar os diagramas, foi utilizada a notação por conjuntos aninhados [Knut73], expressando a hierarquia da árvore através de conteúdo espacial. Cada nó é colocado dentro do seu nó pai. Este paradigma oferece métodos distintos visualmente para exibir a hierarquia e a estrutura de grafos.

Com o objetivo de filtrar algumas informações, o IGD utilizou três técnicas básicas: supressão de detalhes de sub-árvore, seleção de sub-árvore no mapa e herança de elos, que serão discutidas na Seção 3.2.

A herança de elos pode esconder muita informação sendo necessário fornecer alguma maneira de obtê-la. O IGD fornece um recurso para o usuário visualizar todos os elos que saem de determinada página ou capítulo ou os que nela chegam.

## 2.11 CYBERMAP

O CYBERMAP [Gloo91] teve sua implementação feita em um Macintosh utilizando o HyperCard, porém pode complementar os recursos de navegação existentes em um hipertexto qualquer ou servir como ferramenta auto-suficiente para navegação em um documento. Ele introduz o conceito de HYPERDRAWER para particionar os nós em seqüências ordenadas, que será analisado na Seção 3.4.

A idéia do CYBERMAP não é utilizar a estrutura de ligações do documento, mas sim analisar o conteúdo dos nós usando técnicas de indexação e *clustering* automáticas. Baseado nisto, uma nova estrutura de grafo é computada para o diagrama, diferente da estrutura de nós e elos do hipertexto.

## 2.12 Storyspace

O Storyspace [Joyc91] [BBJM91] é um sistema hipermídia da Eastgate Systems que foi projetado como uma ferramenta para o processo de autoria. Ele tem sido amplamente utilizado tanto para escrever livros tradicionais como para fazer hiperdocumentos.

O sistema oferece um espaço de autoria para criação e geração de documentos. Cada idéia toma forma de uma caixa ou célula no espaço de autoria (vide Figura 17).

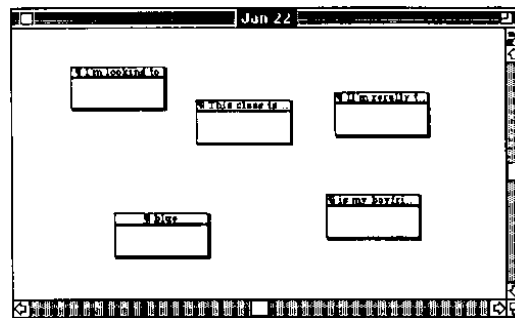


Figura 17: Storyspace - espaço de autoria básico de um autor com 5 células

O usuário pode visualizar a estrutura hierárquica de duas maneiras: através de conteúdo espacial, onde a subordinação é indicada colocando células dentro de outras, ou através de uma árvore ou layout indentado convencional (vide Figura 18).

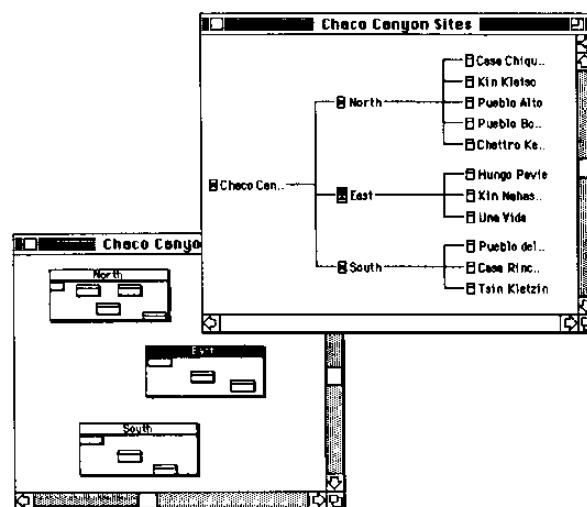


Figura 18: Storyspace - duas visões da mesma estrutura

Para evitar diagramas de árvores muito extensos, o mapa do Storyspace esconde informações sistematicamente. Em uma visão, o sistema mostra somente dois níveis da hierarquia estrutural e somente uma parte do plano de informações. Para acessar as informações escondidas, o usuário deve mudar a visão através de scroll, zoom ou navegando em profundidade pela hierarquia.



Os elos aparecem no mapa como uma seta da célula origem para a célula destino. Se uma das extremidades do elo não está visível, aparece uma seta abreviada na extremidade visível (vide Figura 19).

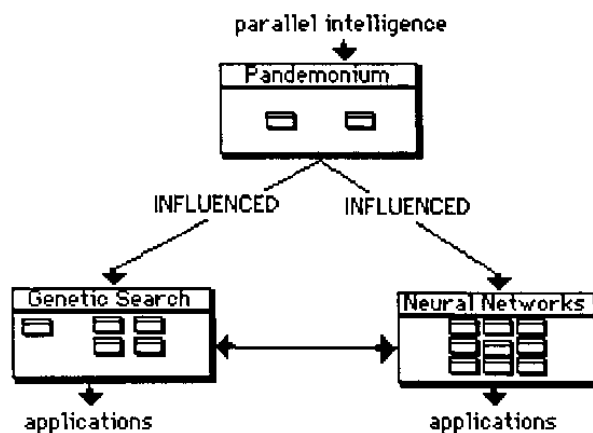


Figura 19: Storyspace - elos

O Storyspace oferece um modo de leitura somente para o browser que é utilizado pelos leitores, onde o usuário só é capaz de acessar as informações e não alterá-las.

O sistema também registra o caminho do usuário ao navegar pelo hiperdocumento e permite a definição prévia de caminhos agrupando uma lista de células.

## 2.13 SHADOCS

SHADOCS (SHARing DOcument System) [ZiLa94] é um protótipo de interface gráfica entre um ou mais usuários e um ou mais espaços de informação. Ele combina técnicas para recuperação de informações com visualização e técnicas de interação para facilitar tarefas como indexação, recuperação e filtragem de informações por um ou mais usuários. SHADOCS utiliza IDM's (Interactive Dynamic Maps) como mecanismo central para suportar essas tarefas e para acessar e navegar no espaço de informações.

A técnica de IDM representa um novo paradigma de navegação baseado na metáfora espacial para ajudar usuários acessarem e naveguem em conjuntos grandes de documentos. Um IDM tem um papel parecido com os mapas reais com relação ao espaço físico. Uma coleção de documentos é considerada um território que contém recursos (documentos e tópicos) e mapas desses territórios podem ser desenhados. Mapas reais tiram vantagem da habilidade do cérebro humano de memorizar e manipular informações espaciais.

Algumas facilidades são oferecidas:

- usuários perdidos sempre acham um mapa mostrando onde estão, de onde vieram e para onde podem ir;
- mapas mudam dinamicamente quando novos documentos são acrescentados à rede;
- mapas suportam navegação, querying e browsing através de manipulação direta;

- mapas podem ser customizados, gravados e reutilizados;
- mapas podem ser compartilhados por um grupo de usuários.

Dois tipos de IDM são computados para os documentos: IDM de tópicos representa o conteúdo semântico de um conjunto de documentos, enquanto IDM de documentos visualiza um subconjunto de documentos, por exemplo os que resultam de pesquisas do usuário.

Foram adaptadas convenções do mundo real sobre o desenho de mapas geográficos para atender às necessidades de exibição da estrutura de hiperdocumentos. Mapas são feitos de regiões, cidades e ruas. Por exemplo, regiões são conjuntos de documentos, o tamanho das regiões reflete o número de documentos na região e a distância entre duas cidades representa as relações de similaridade entre ambas.

IDM de tópicos fornece uma visão global de tópicos representados em coleções, apresentando importância, similaridade e correlações entre eles (vide Figura 20). As áreas do mapa são classes de um *thesaurus* intermediário. O *thesaurus* é um conjunto de tópicos ou descritores que estão correlacionados e organizados em classes. É obtido automaticamente aplicando análises completas de texto a uma rede de documentos e aplicando técnicas de *clustering* ao conjunto de descritores resultante. Criar o *thesaurus* é computacionalmente caro, entretanto, só precisa ser criado uma vez e depois atualizado com novos documentos. No IDM de tópicos, regiões são classes de descritores e cidades são descritores.

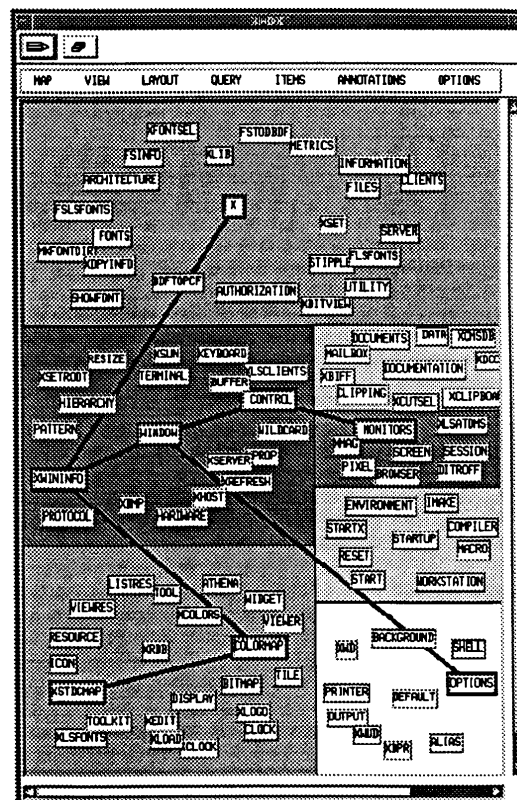


Figura 20: SHADOCS - IDM de tópicos

IDM de documentos representa uma coleção de documentos gerados automaticamente por uma pesquisa do usuário ou agrupados manualmente por usuários ou projetistas. Diferente do IDM de tópicos, as cidades no mapa são documentos (vide Figura 21).

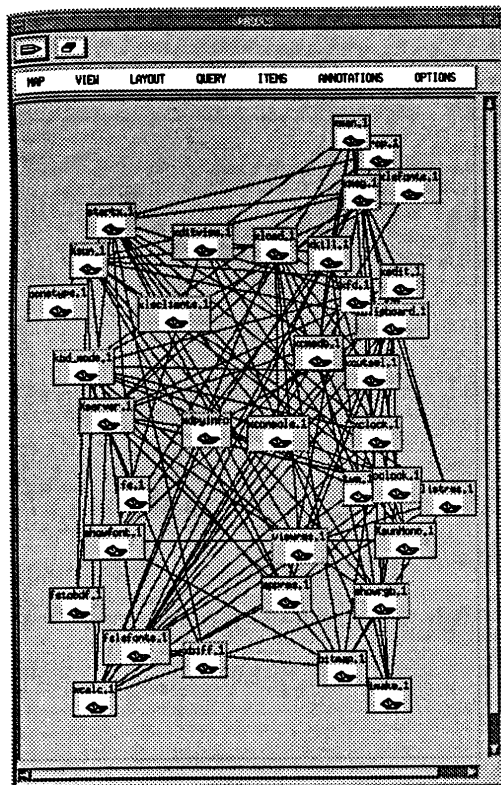


Figura 21: SHADOCS - IDM de documentos

IDM suporta duas técnicas de navegação interativa: browsing ou querying. Browsing permite que usuários explorem um conjunto desconhecido de documentos, onde eles não sabem como descrever o que estão procurando. O sistema identificou três níveis de browsing: browsing de alto nível que utiliza visões olho-de-peixe (vide Seção 3.5), browsing de nível médio que utiliza zoom e browsing de baixo nível que utiliza múltiplas janelas.

Browsing de alto nível é projetado para dar uma visão rápida de rede de documentos. Poucos detalhes são mostrados sobre os itens, exceto para aqueles na área de foco (vide Figura 22).

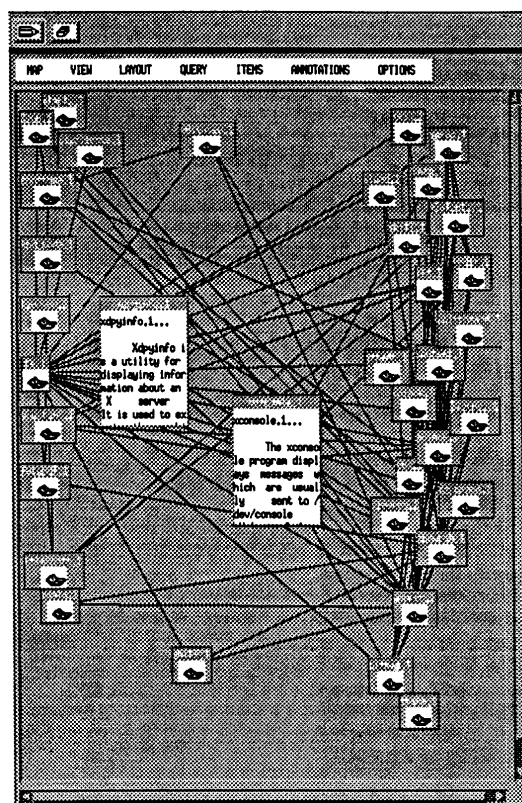


Figura 22: SHADOCS - visão olho-de-peixe

O sistema utiliza customização para implementar mapas ativos. Isto significa que o documento resultante de uma query é recomputado em intervalos de tempo regulares, refletindo as mudanças na rede e filtrando-as.

## 2.14 Analisando os Sistemas Apresentados

Pode-se concluir que é preocupação fundamental dos sistemas hipermídia oferecer aos usuários recursos que auxiliem o processo de navegação e edição de hiperdocumentos. São muitas as tentativas e propostas para apresentar a estrutura da rede de interligações de um documento, sendo que pode-se resumi-las e classificá-las em quatro grupos. Um que apresenta mapas locais com informações relativas à posição do usuário durante o processo de navegação. Outro que apresenta mapas globais que exibem informações de âmbito global sobre todos os documentos existentes. Um terceiro que apresenta mapas globais, porém exibindo informações de uma forma resumida, utilizando algum tipo de técnica de filtragem. E o quarto que apresenta tanto informações locais e globais resumidas em um único mapa.

O sistema Neptune (vide Seção 2.1) oferece apenas um tipo de browser apresentando um mapa global com uma visão ponto-a-ponto do grafo de todos os nós e elos através do seu Browser Gráfico.

O Electronic Document System (vide Seção 2.2) se concentra apenas nos mapas locais centrados na página corrente visualizada pelo leitor e exibindo as páginas diretamente relacionadas a esta.

O NoteCards (vide Seção 2.4) oferece um cartão especial chamado browser que exhibe um diagrama estrutural global da rede de cartões. A chamada History List, extensão do sistema NoteCards já oferece um minibrowser que mostra apenas o cartão corrente e os cartões diretamente relacionados a ele, se enquadrando no grupo de mapas locais.

O Thoth-II (vide Seção 2.5) oferece um mapa que não se enquadra nos quatro grupos de mapas definidos. Este sistema fornece um diagrama atualizado dinamicamente à medida que o usuário navega. Porém, apesar de fornecer um mapa de acordo com a posição do usuário na estrutura, o sistema atualiza o diagrama existente com informações novas sem reiniciá-lo. Portanto, quando o usuário navega por todos os nós do hiperdocumento, o mapa apresentado é similar a um mapa global de toda a estrutura de interligações.

O Intermedia (vide Seção 2.8) é o sistema que mais realizou tentativas de construção de mapas da rede estrutural de informações. Foram construídos mapas globais, mapas locais, mapas com ambas as informações utilizando técnicas de filtragem e chegou-se à conclusão de que apenas os mapas locais e trilhas eram realmente eficazes para o modelo conceitual do sistema que não permite composições aninhadas.

O Storyspace (vide Seção 2.12) oferece mapas globais resumidos só que aplicando um critério de filtragem de informações bastante restrito (são exibidos somente dois níveis de hierarquia) para evitar diagramas muito extensos.

O sistema SHADOCS (vide Seção 2.13), através do IDM de documentos, oferece um mapa com informações globais e locais, utilizando a técnica de filtragem de visões olho-de-peixe, que será abordada com mais detalhes na Seção 3.5. O IDM de tópicos utiliza uma técnica similar a do CYBERMAP (vide Seção 2.11) para computar os mapas. Esta técnica fornece uma visão global resumida sobre os hiperdocumentos, pois não apresenta uma visão ponto-a-ponto da estrutura real dos documentos, e sim de tópicos ou assuntos computados a partir do conteúdo dos documentos através de técnicas de indexação e *clustering*.

Um usuário desorientado precisa tanto de informações locais como de informações de âmbito global para reestabelecer seu senso de localização. Os mapas locais são simples de serem construídos e auxiliam a orientar o usuário que está familiarizado com a estrutura do hiperdocumento. Porém, para aqueles usuários não familiarizados, que não sabem exatamente o item de informação que procuram, somente este tipo de mapa não é suficiente.

Os mapas globais que apresentam uma visão total da estrutura de nós e elos perdem sua utilidade para hiperdocumentos grandes e complexos, pois este diagrama torna-se algo difícil de ser interpretado.

Os mapas globais resumidos são interessantes para fornecer ao usuário uma visão global da estrutura, pois tendem a ser mais legíveis que os mapas globais totais. Estes mapas

combinados com os mapas locais apresentam informações suficientes para o usuário recuperar seu senso de orientação no contexto espacial.

Assim, os mapas que fornecem, em um só diagrama, uma visão global resumida do hiperdocumento e detalhes referentes à posição do usuário em sua estrutura são os mais indicados, pois têm todas as funcionalidades e informações necessárias para ajudar a orientar o usuário espacialmente, respondendo à pergunta “para onde posso ir agora?”.

Outro recurso fundamental para auxiliar o processo de navegação dos usuários é a definição de trilhas, caminhos que podem ser previamente definidos ou registrados durante uma sessão de navegação.

Estudando os sistemas hipermídia, conclui-se que quase todos eles oferecem o recurso de registro do caminho e possibilidade de voltar atrás na trajetória percorrida respondendo à questão dos usuários de “como eu cheguei aqui?”.

Foram mencionados neste capítulo alguns sistemas que utilizam técnicas para melhorar a legibilidade dos diagramas. O sistema GRAB (vide Seção 2.9) apresenta uma melhora no algoritmo de Sugiyama para layout automático de grafos a ser abordado com mais detalhes na Seção 3.1. O sistema IGD (vide Seção 2.10) introduz a idéia de representação das composições através de notação por conjuntos aninhados, extremamente importante para diagramas de estruturas que suportam composições.

Como o modelo conceitual em que esta dissertação se baseia permite composições aninhadas serão utilizadas técnicas de apresentação de diagramas através de conjuntos aninhados como nos sistemas IGD e Storyspace. Entretanto, tem-se como objetivo oferecer aos usuários ferramentas mais flexíveis que permitam moldar a quantidade de informações exibidas no diagrama com a utilização de filtros baseados em visões olho-de-peixe, como no sistema SHADOCS e CYBERMAP. Nestes dois sistemas são utilizadas técnicas para agrupar informações em estruturas que, quando exibidas no diagrama, representam vários nós do documentos em questão. Esta idéia será aproveitada nesta dissertação, só que pelo fato de se utilizar modelos com composições aninhadas de nós, estes agrupamentos serão dados pela própria estrutura do hiperdocumento, não necessitando de processamento para agrupar nós mais relacionados.

A proposta a ser apresentada nesta dissertação visa construir mapas que apresentem informações locais e globais sobre a estrutura de documentos hipermídia e também recursos de navegação por trilhas. Para isso, é necessária uma discussão mais detalhada sobre técnicas de filtragem de informações e de construção de diagramas, que será o assunto do Capítulo 3.