

O DESENVOLVIMENTO DE UM SISTEMA HIPERTEXTO EM AMBIENTE UNIX

Luiz Lima Netto

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

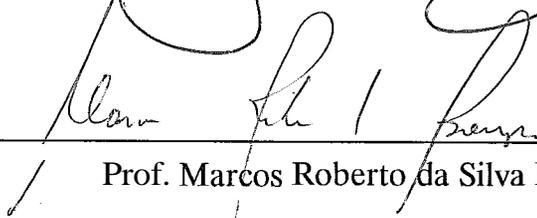
Aprovada por:



Prof. Ronaldo Cesar Marinho Persiano, D. Sc.
(Presidente)



Prof. Jano Moreira de Souza, Ph. D.



Prof. Marcos Roberto da Silva Borges, Ph. D.

RIO DE JANEIRO, RJ - BRASIL
SETEMBRO DE 1991

NETTO, LUIZ LIMA

O Desenvolvimento de um Sistema Hipertexto
em Ambiente UNIX [Rio de Janeiro] 1991.

VIII, 88 p. 29, 7 cm (COPPEIUFRJ, M.Sc.,
Engenharia de Sistemas e Computação, 1991)

Tese - Universidade Federal do Rio de Janeiro,
Coppe

1. Hipertexto

I. COPPEIUFRJ II. Título (série)

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.).

O Desenvolvimento de um Sistema Hipertexto em Ambiente UNIX

Luiz Lima Netto

Setembro de 1991

Orientador: Prof. Ronaldo Cesar Marinho Persiano

Programa: Engenharia de Sistemas e Computação

A finalidade deste trabalho é apresentar o desenvolvimento de um sistema computacional para tratamento de informações que sirva de apoio ao trabalho realizado pelo grupo de interesse em Computação Gráfica da Coordenação dos Programas de Pós-Graduação em Engenharia (COPPE) da Universidade Federal do Rio de Janeiro.

Como fundamento básico no projeto do sistema, foi feita a opção pela exploração do conceito Hipertexto, pois este apresenta, como será visto, alguns aspectos interessantes que parecem ser de uso conveniente neste caso. Além disso, pretende-se também explorar os recursos computacionais disponíveis no ambiente no qual o sistema irá operar.

Este trabalho é dividido em duas partes: a primeira em que é discutido, de modo abrangente, o conceito Hipertexto e a segunda em que é detalhado o projeto do sistema computacional a ser implementado.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.).

The Development of an Hypertext System on the UNIX Environment

Luiz Lima Netto

September, 1991

Thesis Supervisor: Prof. Ronaldo Cesar Marinho Persiano

Department: Engenharia de Sistemas e Computação

The purpose of the present work is the development of a computational system for information management which will support the work done by the interest group on Computer Graphics of this institution.

The Hypertext concept will be the basic framework on the system design because, as will be shown, it presents some interesting aspects that seems to be of convenient use in this situation. As well, computational resources available on the system development environment will be explored.

This document is divided into two parts: a first one where Hypertext aspects are presented and a second one where the system design is detailed.

A Neila e **Auber**, meus pais.

Agradeço ao Prof. Marinho pelos ensinamentos que, muitas vezes, ultrapassavam as fronteiras da Computação Gráfica.

Agradeço ao Prof. Maculan pela oportunidade.

Índice

Parte I – Hipertexto

I – Definindo Hipertexto	02
II – Histórico de Hipertexto	04
III – A Navegação e a Autoria	06
IV – Sistemas Hipertexto	09
V – Algumas Aplicações de Hipertexto	19
VI – Uma Arquitetura para Sistemas Hipertexto	24
VII – Algumas Considerações sobre o Uso de Hipertexto	26

. . .

Parte II – O Projeto

Introdução	30
I – O Levantamento de Requisitos Funcionais	
Introdução	31
I.1 – Tipos de Informação em Sistemas Hipertexto	31
I.2 – Recursos do Processo de Navegação	32
I.3 – Os Requisitos Funcionais	37

II – Aspectos da Interface com o Usuário

Introdução	38
II.1 – A Questão da Usabilidade	38
II.2 – Hipertexto e sua Funcionalidade Básica	39
II.3 – Hipertexto e Usabilidade	42

III – A Proposta de um Modelo

Introdução	4	4
III.1 – O Modelo	44	
III.2 – A Dinâmica do Modelo	47	
III.3 – O Detalhamento do Modelo	49	

IV – A Fase de Implementação 61

IV.1 – Os Módulos de Implementação	61
IV.2 – Considerações sobre a Incorporação de Novos Utilitários	68
IV.3 – O Recurso Comentário	69
IV.4 – Características do Ambiente de Implementação	70
IV.5 – Formatos Padrão do Sistema	71

. . .

Conclusões 73**Bibliografia** 75**Apêndice A: Hyperedit – Tutorial** 80**Apêndice B: Sistema Apoio – Guia de Instalação** 87

Hipertexto

I - Definindo Hipertexto

Existem atualmente na bibliografia diversas descrições de hipertexto. Algumas, mais poéticas, que afirmam: "Hipertexto é liberdade!" ([1]); outras, mais convencionais, definem hipertexto simplesmente como "texto não-linear"([2]). Na verdade, estas definições apresentam as diversas faces de uma mesma idéia que, mesmo singela, pode ser explorada de várias maneiras.

Talvez, o modo mais simples de se entender hipertexto seria imaginar uma série de documentos (textos) que, ao serem interligados, formassem uma rede em que cada nó fosse exatamente um texto (Figura 1). Deste modo, um leitor que quizesse examinar esses textos poderia fazê-lo navegando, de nó em nó, por caminhos livremente escolhidos.

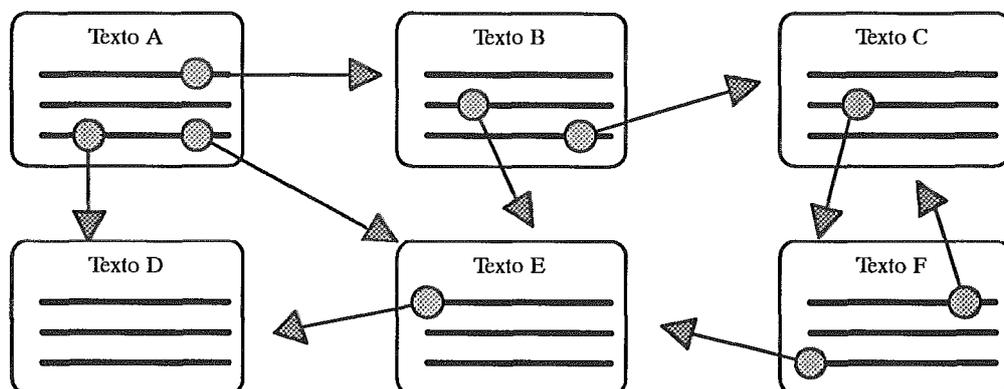


Figura 1 - O conceito de Hipertexto

Em contraposição à idéia de hipertexto, pode-se pensar na disposição linear na qual os capítulos de um livro se encontram. Neste caso, o acesso às informações é normalmente feito de uma só maneira, ou seja, a partir de um ponto qualquer do livro, o leitor pode ir somente para um único outro: a página seguinte. No caso de um hipertexto, o leitor possui, a cada momento, várias opções que podem ser seguidas.

Falando em termos computacionais, um sistema hipertexto deve basicamente prover mecanismos que possibilitem tanto o acesso como a **organização** de informações no modo anteriormente descrito. Além disso, existe uma série de recursos que podem ser incorporados a sistemas desta natureza, oferecendo algumas facilidades adicionais. Em breve, esses recursos serão discutidos.

A operacionalidade de um sistema hipertexto pode se dar de diversas maneiras. Uma delas, a mais comum, é ilustrada na Figura 2 onde um **navegante** que porventura estivesse examinando um texto A, poderia passar a examinar, por exemplo, o texto B ao **apontar** o trecho no texto A que está associado ao texto B.

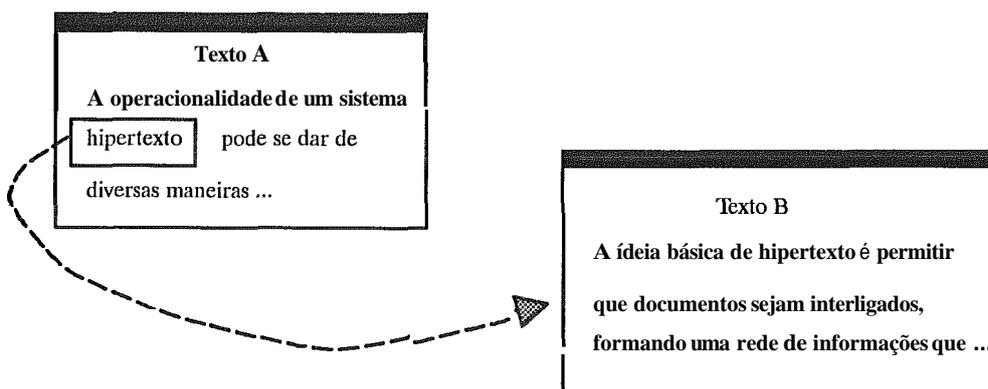


Figura 2 – Hipertexto em uso computacional

Trechos de texto que indicam a associação a outros textos são considerados uma das extremidades de um **elo** de ligação entre dois nós do hipertexto; um que representa o texto de origem e o outro que representa o texto destino. Esses trechos serão chamados, neste trabalho, **chaves de navegação**.

Hipertexto e Hipermídia

Como extensão à idéia inicial de hipertexto em que se considera um estrutura associativa entre documentos textuais, alguns sistemas ([3], [4] etc) permitem a incorporação de outros tipos de informação no hipertexto, como por exemplo: gráficos, imagens, sons ou mesmo animações. Neste caso, são chamados sistemas **hipermídia**.

A possibilidade de tratamento de informações de naturezas diversas é extremamente enriquecedora para um sistema hipertexto, pois há indubitavelmente um poderoso incremento no poder de expressão de idéias, comparado àquele obtido pela utilização exclusiva de textos.

Neste trabalho, será utilizado exclusivamente o termo hipertexto que abrangerá, porém, o sentido da expressão hipermídia.

II - Histórico de Hipertexto

"A mente humana ... opera por associação... O homem não pode pretender duplicar totalmente este processo mental, mas deve ser capaz de aprender com ele... Não se pode esperar igualar a velocidade e a flexibilidade com a qual a mente segue um caminho associativo, mas deve ser possível superar a mente no que concerne o tempo de permanência e a clareza dos itens recuperados do armazenamento."

Vannevar Bush, em 1945.

Considerado como o precursor da idéia de hipertexto, **Vannevar Bush**, cientista e consultor do governo americano, em seu artigo "*As We May Think*" ([5]), descreve um dispositivo capaz de armazenar e consultar de modo rápido e flexível uma vasta gama de documentos científicos que estariam associados, formando uma imensa rede de informações.

"... um dispositivo no qual um indivíduo armazena todos os seus livros, registros e comunicações, sendo mecanizado de maneira tal que possa ser consultado com grande rapidez e flexibilidade. É um suplemento à sua memória."

Por não existir tecnologia adequada na época, **Memex***, que fora idealizado tendo com base microfílmese fotocélulas, nunca foi construído. A idéia, no entanto, estava lançada.

. . .

Como segundo marco no histórico de hipertexto, considera-se o desenvolvimento do sistema **NLS** (On-line System) que integrava o projeto **Augment** (1963) do *Stanford Research Institute* de responsabilidade de **Douglas Engelbart**.

* *memory extender*.

O projeto Augment tinha como objetivo desenvolver ferramentas computacionais que permitissem ampliar a capacidade intelectual humana. Este projeto é considerado pioneiro em áreas como automação de escritórios e processamento de textos.

O sistema NLS, por sua vez, possuía uma série de características de um sistema hipertexto. Foi utilizado, durante o projeto, para gerenciar documentos. Artigos, relatórios e memorandos eram armazenados e havia a possibilidade de se estabelecer referências cruzadas entre os mesmos. Os pesquisadores envolvidos no projeto possuíam, assim, uma ferramenta para o intercâmbio de informações e apoio ao trabalho de pesquisa.

. . .

Foi somente em 1967 que o termo hipertexto foi citado pela primeira vez. **Ted Nelson**, ao idealizar o sistema **Xanadu** ([6] citado em [2]), desejava interligar todas (!) as referências literárias existentes na face da Terra, formando um hipertexto verdadeiramente universal que pudesse ser acessado de modo interativo.

O sistema Xanadu foi parcialmente desenvolvido, mas obviamente (?) não atingiu os objetivos iniciais. Uma versão se encontra disponível comercialmente.

III - A Navegação e a Autoria

De modo geral, sistemas hipertexto suportam duas atividades básicas: a **navegação** e a **autoria**. A primeira consiste em percorrer a estrutura do hipertexto e a segunda consiste em compô-la. Dependendo da situação, pode haver ou não uma separação entre os usuários que realizam essas duas tarefas, ou seja, em alguns casos, existe uma distinção clara entre autores e navegadores.

O Processo de Autoria

No caso específico do processo de autoria, existe uma série de questões a serem consideradas que implicam em tomadas de decisão sobre a organização da estrutura no hipertexto como um todo e sobre como compor cada documento individualmente.

Em relação à composição individual, considera-se como boa estratégia a elaboração de documentos em tamanho moderado e não muito abrangentes, restritos a um tópico. Na estruturação, por sua vez, somente as associações mais relevantes e realmente importantes devem ser consideradas. A idéia básica é simplificar a tarefa de compreensão dos documentos e evitar que os navegantes se desgastem ao terem que constantemente decidir se devem ou não seguir pelos caminhos possíveis.

Viabilidade de elaboração de hipertextos adequados está intimamente ligada a recursos que o sistema oferece. Falando em termos operacionais, por exemplo, é conveniente que se tenha, entre outras coisas, facilidades na especificação de associações, disponibilidade de uso de ferramentas poderosas (editores de texto, editores gráficos* etc) e facilidades para consultas a bases de informação.

Uma idéia que merece destaque, relacionada ao processo de autoria, é aquela referente ao **trabalho cooperativo**. Neste, autores diversos, mesmo distantes fisicamente, podem contribuir na elaboração do hipertexto. É certamente um poderoso instrumento para apoio a grupos de trabalho.

Não se pode falar, até hoje, em uma exata descrição do processo de autoria. Na verdade, do mesmo modo como algum tempo é levado para se aprender a criar textos lineares, algum tempo se dará até que se tenha uma idéia clara de como elaborar hipertextos. Por enquanto, esta tarefa fica a cargo do bom senso e da experiência de cada autor.

*partindo do pressuposto que o sistema apresente o potencial de manipulação de gráficos ou imagens.

O Processo de Navegação

É no processo de navegação que se resgata a real essência da idéia de hipertexto. Nesse, o navegante percorre interativamente a estrutura do hipertexto, acessando os documentos interligados.

Em geral, o objetivo da navegação é a recuperação de alguma informação, porém, em alguns casos específicos como, por exemplo, o do uso de hipertexto na Área Educacional, o próprio ato de navegar pode assumir um papel relevante (veja subseção Hipertexto em Aplicações Educacionais, página 22).

Para ilustrar o que seria um processo de busca de informações em um sistema hipertexto, pode-se considerar o exemplo em que um pesquisador se dirige a uma biblioteca para efetuar uma pesquisa bibliográfica sobre aves marinhas. Nesta biblioteca, automatizada, o pesquisador encontra um sistema hipertexto que mantém uma rede de informações formada pelas descrições de todos os livros e artigos disponíveis sobre o assunto. Cada componente do hipertexto descreve, portanto, uma obra e referencia outras (Figura 3).

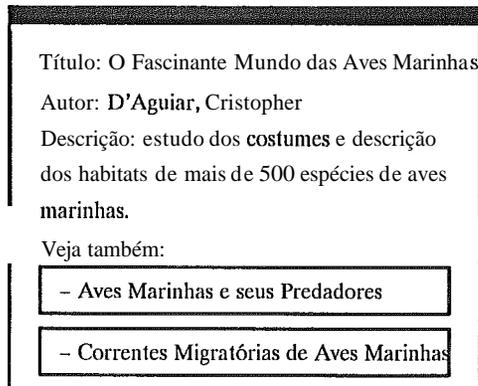


Figura 3 – Hipertexto Bibliográfico

A pesquisa por esta "rede bibliográfica" consistiria em um contínuo processo de exames sucessivos de descrições de obras que estivessem associadas. Deste modo, durante a navegação, o pesquisador poderia coletar referências de seu interesse.

Não é difícil imaginar que, devido à grande quantidade de obras existentes na biblioteca, o pesquisador do exemplo anterior poderia ter, em algumas situações, dificuldades para encontrar alguma referência específica através do sistema hipertexto. Por esta razão, alguns sistemas oferecem facilidades (mecanismos de indexação, busca por palavras chave etc) que auxiliam na busca de informações.

A simplicidade operacional do processo de navegação é um dos requisitos que um sistema hipertexto deve oferecer, pois, deste modo, garante-se a praticidade no uso do mesmo. Entre outras, esta é uma das razões pela qual a elaboração da interface com o usuário de um sistema desta natureza é de fundamental importância.

Um ponto interessante que deve ser destacado refere-se à relação existente entre o processo de autoria e o que pode ser chamado de **navegabilidade** do hipertexto durante o processo de navegação. A navegabilidade refere-se à capacidade de controle do processo durante a navegação que está intimamente relacionada à estruturação do hipertexto, realizada durante a autoria.

IV – Sistemas Hipertexto

Os anos 80, ou seja, algum tempo depois do desenvolvimento dos sistemas NLS (1968) e Xanadu (1967), considerados como os primeiros sistemas hipertexto a serem implementados, foram palco do surgimento de diversos outros sistemas. Todos estes, embora possuíssem características específicas, eram também baseados na estruturação associativa entre informações. Nesta seção, serão apresentados alguns desses sistemas hipertexto, sendo visto também algumas extensões ao conceito básico de hipertexto que foram introduzidas pelos mesmos.

Intermedia

Intermedia ([3] e [7]) é um sistema que vem sendo desenvolvido por um grupo pertencente ao Institute *for* Research in *Information* and Scholarship (IRIS) da Universidade de Brown. Este grupo é considerado como um dos mais experientes no estudo de hipertexto, tendo sido responsável por projetos pioneiros na área.

O sistema Intermedia permite a manipulação, não só de textos, como também de gráficos ou imagens. Além disso, foi projetado para ser um ambiente integrado no qual vários tipos de aplicações pudessem ser interligadas (Figura 4).

Um conceito interessante introduzido no sistema Intermedia é o de teia (web). A idéia é fazer com que somente um número restrito de nós, ou seja, aqueles pertencentes à teia que esteja aberta, sejam apresentados ao usuário a cada momento.

A possibilidade de seleção de teias durante a navegação permite que esta se torne contextual, pois somente um grupo específicos de nós estará, em função do contexto, ao alcance do navegante. Estes mesmos nós podem, por sua vez, pertencer a mais de uma teia.

O sistema Intermedia foi idealizado para apoio educacional. Com ele, professores organizam e apresentam material didático a seus alunos. Estes últimos utilizam o sistema para estudo e também como fonte de consulta em trabalhos escolares.

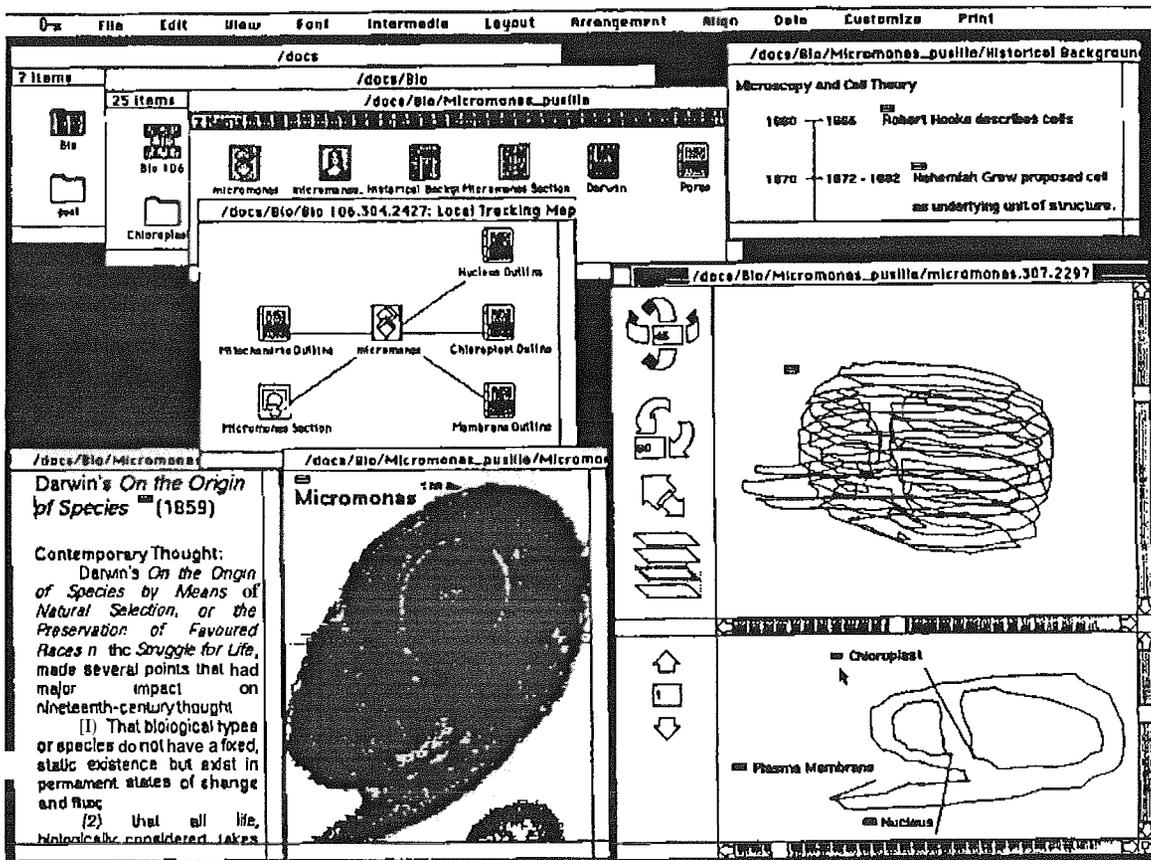


Figura 4 - O Sistema Intermedia

NoteCards

NoteCards ([8]) é um sistema que foi desenvolvido em *Palo Alto Research Center* (PARC), pertencente a empresa Xerox. É considerado um dos sistemas hipertexto mais conhecidos.

Quatro elementos básicos são definidos no sistema NoteCards: *notecards*, *fileboxes*, *links* e *browsers*.

. cada *notecard* representa um nó do hipertexto e está associado a uma janela na tela para apresentação de informações. Estas informações podem ser de naturezas diversas (textos, gráficos etc). Além disso, usuários podem especificar uma estrutura interna para um grupo de *notecards*, definindo, na verdade, **tipos de notecards** no sistema;

. *filebox* é um tipo específico de *notecard* que serve para agrupamento de *notecards* ou também de outros *fileboxes*, possibilitando a criação de estruturas hierárquicas;

. *links* são elos de ligação entre *notecards* (ou *fileboxes*). Mais uma vez, usuários podem definir tipos *de links* ao especificar, através de um rótulo, as relações existentes entre nós origem e nós destino;

. o último elemento, browser, apresenta diagramas estruturais do hipertexto. Com estes diagramas, além de navegar, os usuários podem fazer alterações na estrutura e no conteúdo do hipertexto.

O sistema NoteCards foi inicialmente concebido para auxílio ao trabalho intelectual de análise de informações, ao permitir que grupos de idéias interrelacionadas pudessem ser apresentadas através de uma rede de informações.

KMS

KMS (*Knowledge Management System*) ([9]) é reconhecidamente um dos mais antigos sistemas hipertexto em uso, pois é derivado do projeto do sistema ZOG ([10] citado em [9]) que teve início em meados dos anos 70 e que fora desenvolvido na Universidade de Carnegie Mellon.

O modelo utilizado no sistema KMS pode ser visto na Figura 5. Neste, o elemento básico, chamado de *frame*, pode conter textos, gráficos ou imagens digitalizadas. Além disso, cada frame é composto também por um nome, um título, ítems para associações hierárquicas, ítems de comando e ítems de anotação.

As associações entre *frames* são geralmente especificadas de modo hierárquico através dos ítems para associações hierárquicas, porém *frames* que não estejam subordinados em uma mesma hierarquia podem ser também interligados através dos ítems de anotação que possibilitam, entre outras, esta operação (Figura 6).

Uma das características mais interessantes do sistema KMS é que ele permite o trabalho cooperativo entre autores, possibilitando a distribuição das informações utilizadas em diversos pontos de uma rede de computadores. Além disso, não é feita nenhuma distinção entre autores e navegadores, pois qualquer um que esteja utilizando o sistema pode efetuar alterações nos dados que estão sendo acessados.

Um último ponto a destacar sobre o sistema KMS é a não utilização de janelas na apresentação de *frames*; em geral, a tela é dividida em duas partes e em cada uma destas, um *frame* é exibido. A interação usuário/sistema é feita através de manipulação direta com o uso de cursor/ratinho.

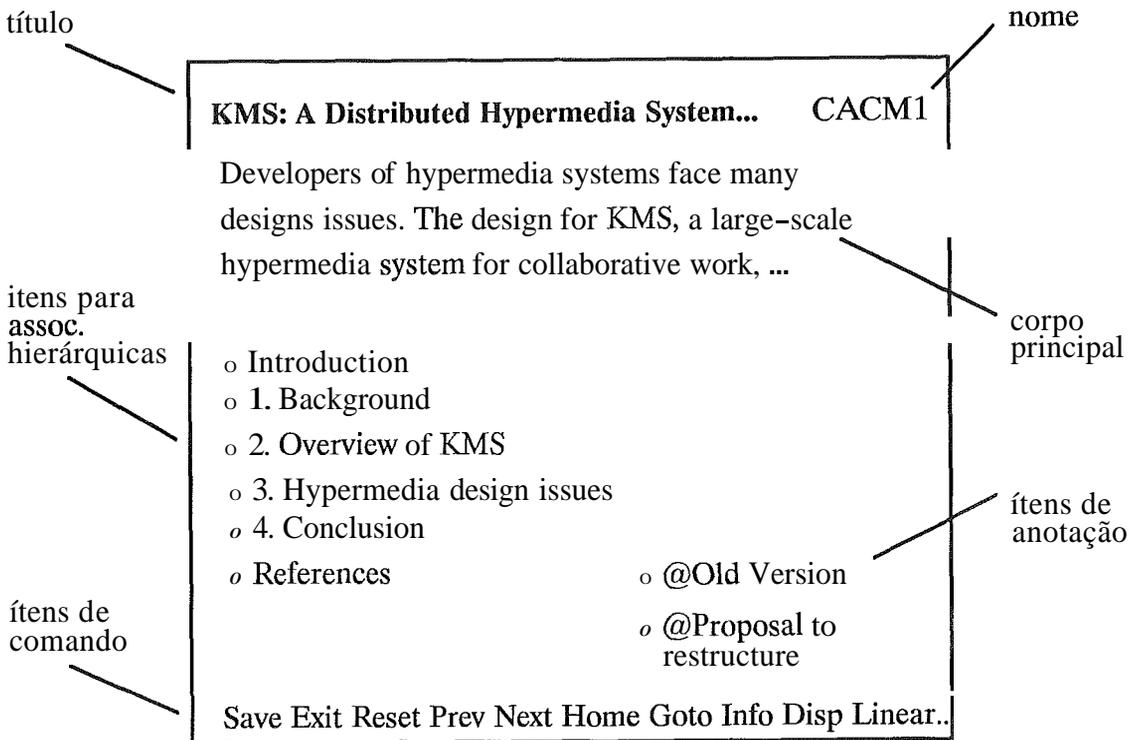


Figura 5 - *Frames* : o modelo no sistema KMS

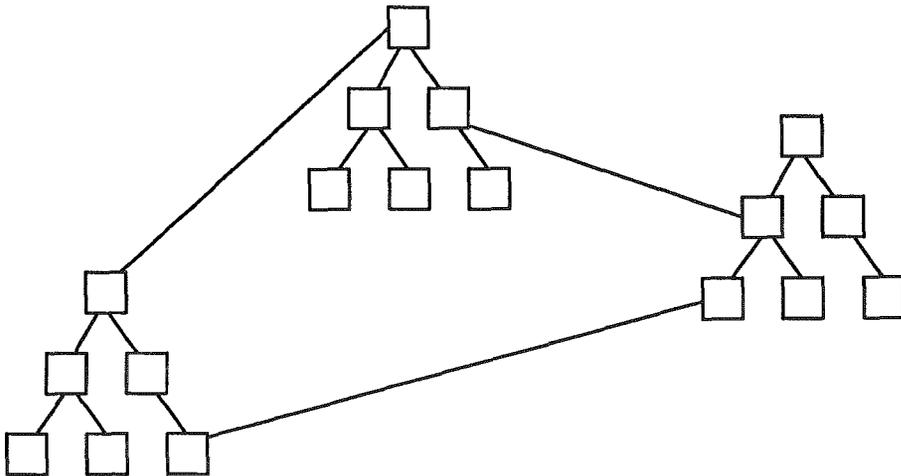


Figura 6 - Associações inter-hierárquicas de *Frames*

Até hoje, o sistema KMS foi utilizado em um grande número de aplicações como: gerenciamento de projetos, documentação on-line, engenharia de software e CAI (*Computer-Aided Instruction*).

Hiperties

O sistema Hiperties ([11]) é fruto de um projeto que vem sendo desenvolvido pela Universidade de Maryland desde 1983. **As** primeiras versões do sistema operavam exclusivamente em microcomputadores pessoais; atualmente, existem versões que operam em redes de estações gráficas.

As unidades básicas de informação no sistema Hiperties* são chamadas **artigos** que são interligados através de **elos**. Estes últimos são representados, em cada artigo, por intermédio de palavras ou frases que são exibidas na tela com realce em brilho (Figura 7). O sistema oferece também recursos para apresentação de imagens.

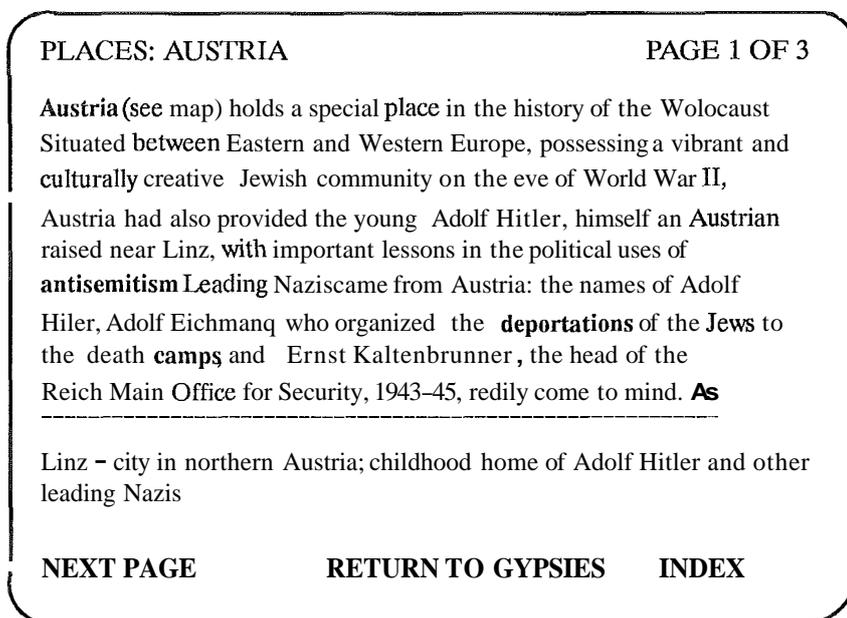


Figura 7 - Uma tela típica do Sistema Hiperties

As técnicas de interação do sistema Hiperties são extremamente singelas, permitindo que usuários sem experiência computacional utilizem o sistema. Durante a apresentação de artigos, por exemplo, a ativação de elos pode ser feita com uma operação simples através do teclado.

*ties vem de *The Electronic Encyclopedia System*.

De modo análogo ao utilizado pelo sistema KMS, a área inteira da tela é utilizada na apresentação de artigos, ou melhor, na exibição, uma a uma, das páginas que os compõem. Não há, portanto, o emprego de janelas.

O sistema Hiperties oferece alguns recursos de apoio à navegação: **histórico***, **índice de títulos** de artigos e exibição de **dicas***² entre a apresentação de dois artigos sucessivos.

Hiperties foi utilizado em vários tipos de aplicações, como por exemplo: sistema de informações para museus, sistema de informações bibliográficas e sistema de apoio a ensino. Demonstra, com isso, a sua versatilidade.

Neptune

Neptune ([12]) é um sistema hipertexto desenvolvido em *Tektronix Computer Research Laboratory* que vem sendo utilizado como suporte a projetos na área de Engenharia de Software. Sua arquitetura é baseada em camadas, existindo uma separação clara entre o nível que trata da interface com o usuário e o nível que oferece recursos de hipertexto, chamado *Hypertext Abstract Machine* (veja Figura 14, página 24).

O nível HAM (*Hypertext Abstract Machine*) oferece operações básicas para criação, modificação e acesso a nós e elos, que podem ser associados a pares do tipo **atributo/valor**. Este último recurso permite também a busca de informações em hipertextos através de um mecanismo de **filtragem**.

O sistema Neptune (Figura 8) possibilita a distribuição de informações através de redes de computadores com controle de acesso simultâneo de usuários. Além disso, possui um esquema para tratamento de **versões** de nós e elos. Com este último, é possível manter várias versões de um hipertexto, sendo esta uma necessidade evidente quando se pensa em controlar projetos de desenvolvimento de software.

A arquitetura apresentada pelo sistema Neptune permite que vários tipos de aplicações sejam suportadas através de alterações específicas no nível de interface com o usuário. Devido a esta flexibilidade, o modelo HAM vem despertando interesse em vários grupos de pesquisa e, por isso, voltará a ser comentado em uma próxima seção deste trabalho.

*1 histórico permite que o usuário veja a sequência em que os artigos foram acessados.

*2 uma dica descreve sucintamente o conteúdo do artigo que vai ser acessado.

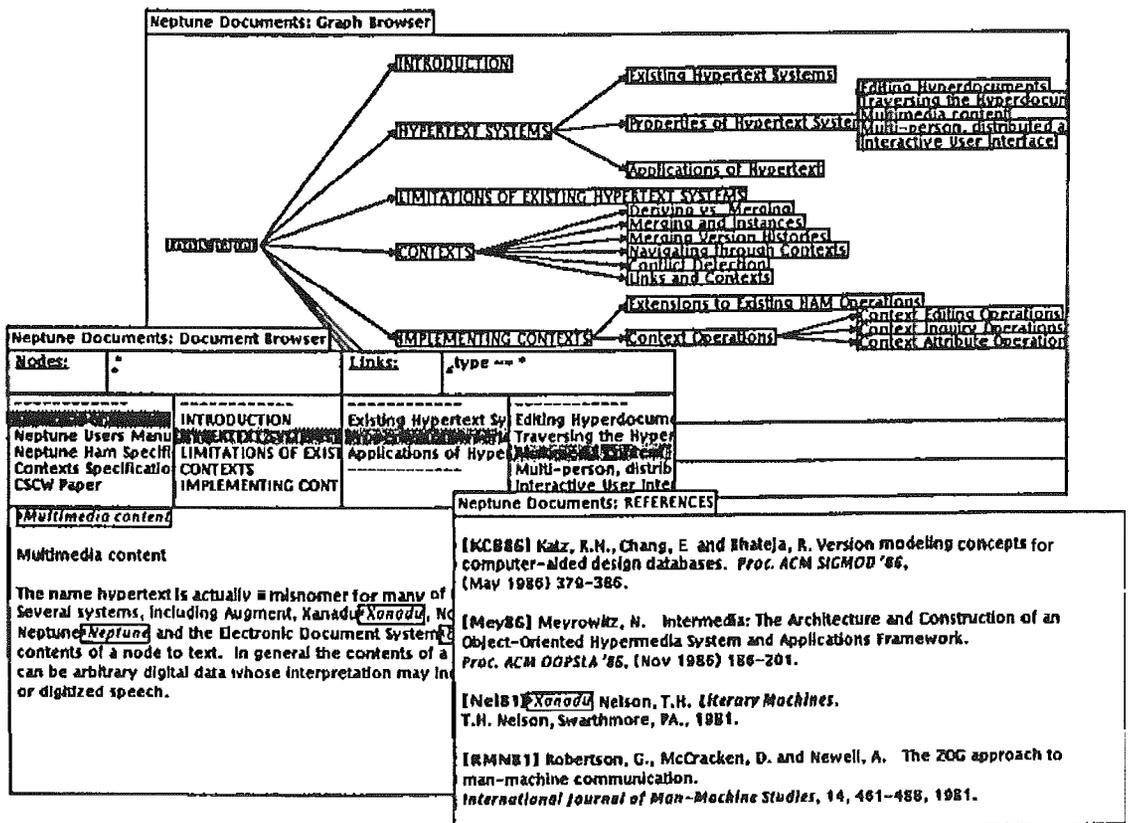


Figura 8 – O Sistema Neptune

Hypercard

Hypercard ([4] e [13]) é considerado o sistema que mais contribuiu para a disseminação do emprego de hipertexto em aplicações computacionais. Foi desenvolvido pela empresa Apple e se encontra disponível desde 1987.

O sistema Hypercard (Figura 9) apresenta o poderoso recurso de **execução de scripts**. *Scripts* são pequenos programas escritos em uma linguagem de programação, chamada Hypertalk, que foi especialmente desenvolvida para que usuários com pouca experiência computacional pudessem utilizá-la. Mesmo assim, a execução de *scripts* permite a realização de uma série de operações que vão desde a apresentação de uma sequência sonora até a execução de outros programas.

Os elementos básicos de informação no sistema Hypercard são chamados de **cartões (cards)** que podem ser organizados em estruturas chamadas de **pilhas (stacks)**. Em cada cartão, estão definidos uma série de **botões (buttons)**. Estes, ao serem ativados, executam um *script* que pode, por exemplo, determinar a apresentação de um novo cartão. É deste modo que elos entre cartões são estabelecidos.

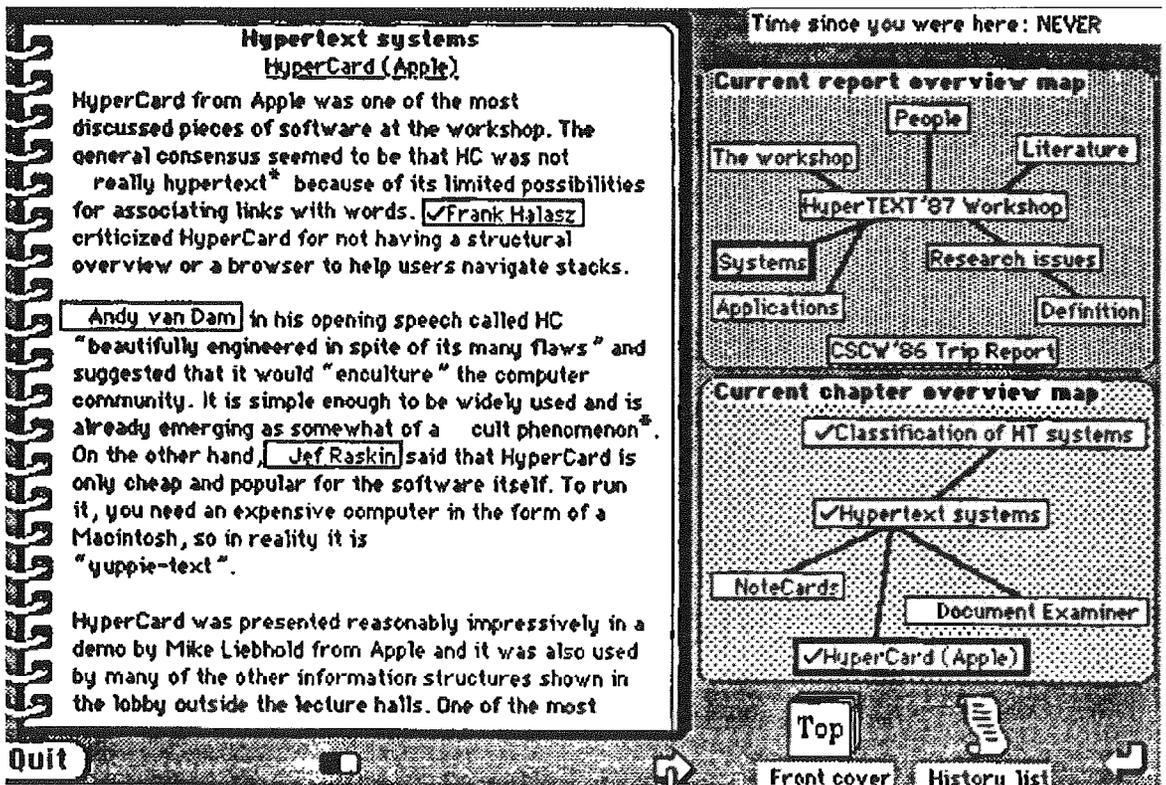


Figura 9 – O Sistema Hypercard

Durante suas apresentações, os cartões ocupam toda a área da tela do mesmo modo como ocorre nos sistemas KMS e Hiperties. Além disso, o sistema Hypercard oferece uma série de facilidades para tratamento de informações gráficas que permitem sofisticadas a nível de interface com o usuário.

O sistema Hypercard vem sendo utilizado em uma série de aplicações. Devido, principalmente, à presença de recursos gráficos poderosos e à flexibilidade da linguagem Hypertalk, ele vem demonstrando uma grande aceitação por usuários em geral.

Guide

O sistema Guide ([14]) foi desenvolvido a partir de um projeto que se iniciou na Universidade de Kent em 1982. Dentre os sistemas hipertexto disponíveis para microcomputadores pessoais, Guide é um dos mais populares.

As unidades básicas de informação no sistema Guide são **documentos** com informação textual. Em relação a elos, o sistema suporta basicamente três tipos: elos de substituição (replacement-buttons), elos de anotação (note-buttons) e elos de referência (reference-buttons).

Elos de substituição podem ser utilizados, por exemplo, para interligar seções e subseções de um documento em um enfoque de estruturação hierárquica. Quando um elo de substituição é ativado o texto referente ao elo é expandido, sendo substituído pelo texto destino (Figura 10).

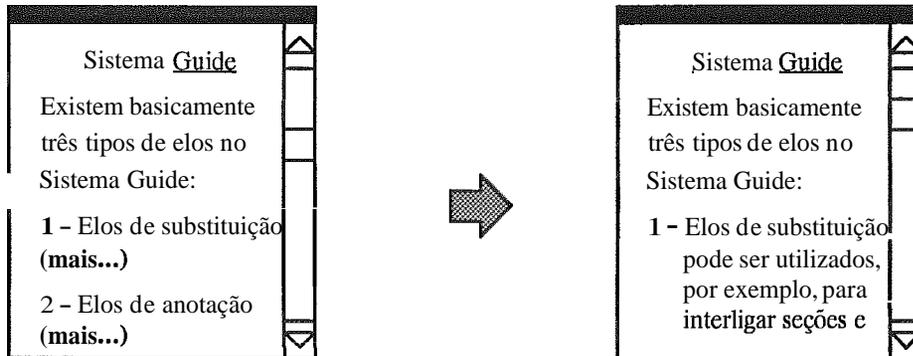


Figura 10 – A expansão de texto através de elo de substituição

Elos de anotação são utilizados para exibir pequenas notas* de texto que estejam associadas ao documento principal. Isto é feito através de pequenas janelas que são abertas na tela.

Elos de referência permitem que documentos distintos sejam interligados. A ativação de um destes elos pode acarretar também o desvio para um outro ponto dentro do mesmo documento.

Um aspecto interessante em relação à interface do sistema Guide é o modo pelo qual os usuários ficam sabendo os tipos dos elos que estão distribuídos por um documento. Para isto, eles movem o cursor por cima das regiões que indicam a presença de elos e, a partir da forma que este toma, o tipo do elo em questão fica revelado (Figura 11).

* como esta, por exemplo.

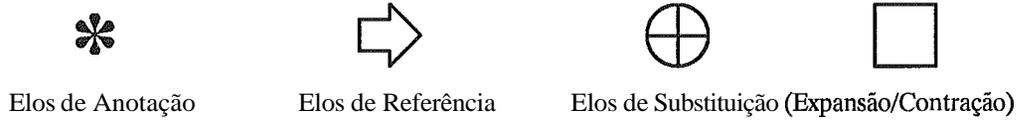


Figura 11 - Formatos de cursor para diferentes tipos de elo

V – Algumas Aplicações de Hipertexto

Não é difícil imaginar a abrangência existente no emprego de hipertexto; basta constatar que diversos *universos de informação* podem ser vistos como um conjunto de associações entre unidades de informação, ou seja, sobre o prisma de hipertexto.

Nesta seção, serão apresentadas algumas possíveis aplicações de hipertexto que, em alguns casos, podem ser suportadas por sistemas desenvolvidos especificamente para elas e, em outros, por sistemas hipertexto de propósito geral. Pretende-se, assim, dar uma idéia geral da flexibilidade no uso deste conceito.

Dicionários e Enciclopédias

Dicionários e enciclopédias são considerados bons candidatos a se tornarem aplicações hipertexto, pois são compostos originariamente por uma série de referências cruzadas que são efetivamente utilizadas durante consultas comuns.

Em um dicionário, por exemplo, quando se busca o significado de uma palavra, é usual que outros termos referenciados sejam examinados pelo leitor, caracterizando um verdadeiro processo de navegação pela obra. A automatização deste mecanismo, além de oferecer maior conforto aos próprios leitores, poderia permitir sofisticações como, por exemplo, a de se poder verificar, através de recursos sonoros, a pronúncia das palavras catalogadas.

De modo análogo aos dicionários, enciclopédias, que, em muitos casos, são formadas por dezenas de volumes, possuem diversas referências internas; de modo distinto, porém, enciclopédias são também compostas por ilustrações e dão margem à exploração de diversos outros tipos de recursos computacionais como a exibição de sequências de vídeos.

Um exemplo de conversão de dicionário para a forma hipertexto vem sendo realizado com o *Oxford English Dictionary* ([15]) que consiste de 12 volumes com mais de 250 mil entradas e 500 mil referências e é considerado um dos maiores dicionários de inglês existentes até então.

Sistemas de Ajuda e Manuais Interativos

A leitura de manuais é considerada uma das tarefas mais ingratas por pessoas que frequentam ambientes computacionais, de modo que, em muitos casos, ela só ocorre quando situações realmente críticas são atingidas. Deste modo, elementos de software (pacotes, bibliotecas, sistemas etc) já podem ser vistos sendo distribuídos com manuais e sistemas de ajuda interativos na forma de hipertexto. Estes são utilizados para se obter alguma informação no momento exato em que esta se torna necessária.

A forma de hipertexto para sistemas de ajuda e manuais interativos permite que várias informações correlatas permaneçam associadas e possam, a critério do usuário, ser acessadas. Em algumas situações de erro, por exemplo, é comum a necessidade de acesso a diversas informações de modo que esta possa ser contornada.

Manuais interativos na forma de hipertexto vêm sendo utilizados, não somente em aplicações computacionais, mas também já acompanham produtos industriais e auxiliam na realização de tarefas como o conserto de automóveis e a manutenção de peças de aviões ou navios (p. ex.: [16]). Em alguns casos, sequências de vídeo são introduzidas nos manuais, detalhando operações que são de difícil descrição.

Uma outra vantagem do uso de um hipertexto automatizado na apresentação de um manual pode ser chamada de **reconfiguração dinâmica** do mesmo. Esta refere-se à possibilidade de um mesmo documento ser apresentado de forma distinta em função, por exemplo, da experiência do usuário que está correntemente utilizando-o. Deste modo, tomando-se o exemplo de um manual de reparo de automóveis, um técnico com pouca experiência poderia, com maiores detalhes, obter os passos para a execução de uma tarefa. Obviamente, este tipo de facilidade pode ser oferecida mais facilmente por um sistema automatizado.

Hipertexto e Engenharia de Software

A Engenharia de Software é uma disciplina em que um dos objetivos primordiais é controlar e aprimorar o difícil processo de desenvolvimento de software. Para tal, são empregadas metodologias, suportadas muitas vezes por conjuntos integrados de ferramentas automatizadas nos chamados ambientes CASE (*Computer-Aided Software Engineering*).

Especificações, documentação de programas, planos de teste e manuais para usuários são alguns exemplos dos diversos documentos que são gerados durante o desenvolvimento de um software. Todos esses estão associados de modo que alterações realizadas em um deles devem ser propagadas por diversos outros documentos com o objetivo de se manter a **consistência** geral entre os mesmos (Figura 12). Nesse contexto, um sistema hipertexto pode ser utilizado para automatizar esta tarefa que, devido à alta dinamicidade do processo de desenvolvimento de software, é realizada em diversos momentos.

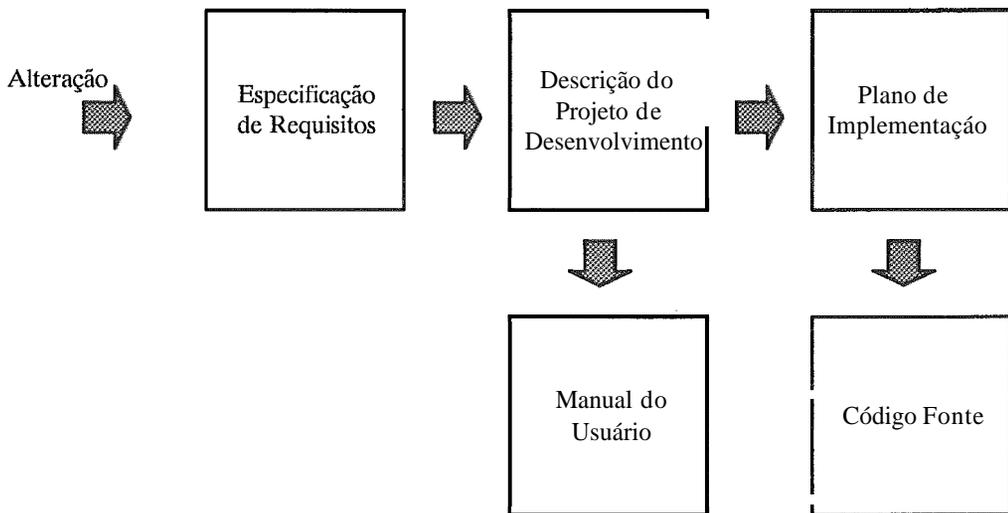


Figura 12 – Propagação de alterações em documentos

Um sistema hipertexto para ambientes de desenvolvimento de software poderia, portanto, auxiliar no apoio àquela metodologia que se fizesse necessária. Vale lembrar que, nesses ambientes, é comum o envolvimento de um grande número de pessoas em tarefas interligadas. Deste modo, o suporte ao **trabalho cooperativo** pelo sistema hipertexto seria conveniente.

O **controle de versões** de documentos é um outro recurso que pode ser considerado quase como essencial em um sistema hipertexto para ambientes de desenvolvimento de software. Históricos de alterações em especificações, por exemplo, são de extrema utilidade e podem ser implementados através daquele mecanismo.

Um exemplo de ambiente CASE que se baseia nos conceitos de hipertexto é o chamado **Dinamic Design** ([17] e [18]) (Figura 13) que apresenta, entre outros, o recurso de controle de versões, citado anteriormente.

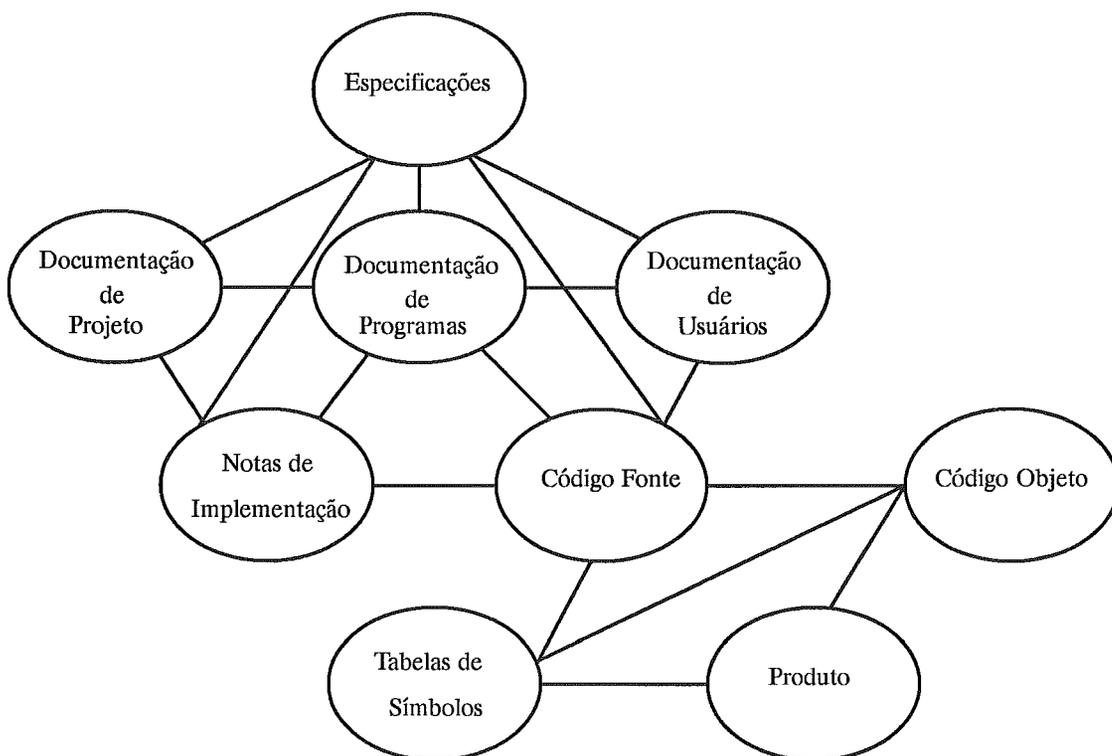


Figura 13 – Dinamic Design: interconecções entre os elementos do ambiente

Por último, vale lembrar que, ainda se referindo a ambientes de desenvolvimento de software, existe uma outra modalidade de sistema hipertexto que pode e vem sendo utilizada: são os chamados sistemas para **organização de idéias** (p. ex.: [19]) que são projetados para o auxílio à tarefa de avaliação de situações complexas, muito comuns em tais ambientes.

Hipertexto em Aplicações Educacionais

A utilização de computadores na área educacional, em especial, em ambientes de CAI (*Computer-Aided Instruction*) vem, cada vez mais, se difundindo nos dias atuais ([20]). Neste quadro, hipertexto pode ser considerado como um enfoque de emprego promissor, pois ele permite a construção e o percurso em modelos conceituais de espaços de informação baseados em associações entre idéias. Deste modo, falando-se em termos do que se pode chamar de **processo de aprendizado** ([21]), pode-se esperar o aprimoramento de tarefas como a memorização, a formação de conceitos e a compreensão. Vale lembrar que o precursor da idéia de hipertexto, Vannevar Bush (veja capítulo II – Histórico de Hipertexto, página 4), menciona, em seu artigo "*As We May Think*" ([5]), que "a mente humana ... opera por associação...".

Além da própria motivação acarretada pelo simples uso de um computador, o emprego, em particular, de um sistema hipertexto pode ser especialmente estimulante de um outro fator importante em ambientes de aprendizado: a iniciativa dos aprendizes. Isto se dá devido à facilidade operacional, quase intuitiva, que sistemas hipertexto, em geral, apresentam na busca de informações. Este é um dos motivos pelos quais o projeto da interface com o usuário para um sistema desta natureza é de extrema importância.

Um outro ponto a se considerar em relação aos recursos que um sistema hipertexto para ambientes educacionais deve oferecer refere-se à possibilidade de apresentação de informações gráficas e/ou dinâmicas (vídeos, animações etc), pois estas permitem um aumento considerável no poder de expressão de idéias, comparado àquele obtido pela utilização exclusiva de informações textuais. Além disso, deve-se considerar a conveniência do sistema suportar facilidades que possibilitem atividades de integração no ambiente como, por exemplo, trabalho em grupo com troca de opiniões.

Já tendo sido utilizado em paralelo a cursos de Literatura Inglesa e Biologia, entre outros, o projeto Intermedia ([3] e [4]) (veja subseção Intermedia, página 9), da Universidade de Brown, é um exemplo de sistema hipertexto que foi idealizado para apoio a ambientes educacionais.

VI – Uma Arquitetura para Sistemas Hipertexto

Na Figura 14, é apresentada uma arquitetura para sistemas hipertexto que é dividida em três níveis:

- . Nível de Apresentação;
- . *Hypertext Abstract Machine* (HAM);
- . Nível de Base de Dados.

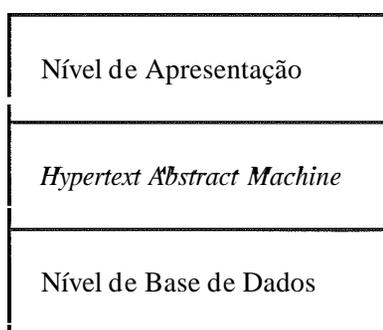


Figura 14 – A arquitetura HAM

Esta arquitetura, apresentada por GAMPBELL e GOODMAN* ([22]), foi utilizada no desenvolvimento do sistema Neptune (veja subseção Neptune, página 14) e visa a formalização de um modelo para sistemas hipertexto que seja ao mesmo tempo flexível e abrangente. A idéia é permitir que sistemas hipertexto que atendam a aplicações diversificadas possam ser estruturados de maneira semelhante. Neste caso, o nível intermediário, HAM, seria responsável pelo "suporte conceitual" aos fundamentos de hipertexto.

O Nível de Base de Dados

O nível mais baixo, o de Base de Dados, é responsável por questões relativas ao armazenamento e à recuperação de informações genéricas, ou seja, por questões que não estão relacionadas especificamente a hipertexto. Neste nível, nós e elos, por exemplo, não são reconhecidos como entidades estruturais de hipertextos.

*nesta seção, não será discutido minuciosamente o modelo de CAMPBELL/GOODMAN. Somente os fundamentos básicos serão apresentados como proposta de generalização.

Deste modo, os níveis superiores do sistema não tem conhecimento nem de como e nem de onde os dados estão armazenados. Pode-se até pensar na distribuição das informações por uma rede de computadores, estando essas armazenadas em uma diversa gama de dispositivos (discos rígidos, CDs etc). O controle ficaria a cargo do nível de Base de Dados. **Além** disso, questões tradicionais como a de restrições de acesso, de compartilhamento de dados e de segurança, seriam também tratadas por este nível.

O Nível HAM

Existe uma série de derivações que, a partir da idéia básica de nós e elos, podem ser consideradas no projeto de um sistema hipertexto. Elos ou nós tipados, subredes, especificação de atributos, são alguns exemplos. Estas derivações objetivam, entre outras coisas, sofisticar o poder de expressão de hipertexto.

O nível HAM é responsável por oferecer recursos associados a um modelo genérico de hipertexto que, baseado na idéia básica de nós e elos, permita a idealização de estruturas gerais, além de possíveis derivações.

Levando-se em conta que o nível de Base de Dados é fortemente dependente de características físicas do ambiente de operação de um sistema hipertexto e que o nível de Apresentação é, como será visto, fortemente dependente do tipo de aplicação do sistema, o nível HAM deve ser inicialmente considerado ao se pensar em padronização de características de sistemas hipertexto que começa a ser uma preocupação emergente nos dias atuais ([23]).

O Nível de Apresentação

É através do nível de Apresentação que as informações do hipertexto são exibidas e a interação do sistema com os usuários ocorre. Dependendo do tipo de aplicação, os modos de apresentação e interação podem diferir, ou seja, a interface com o usuário do sistema está intimamente relacionada às características do ambiente de operação do mesmo, onde se leva em consideração o perfil dos usuários e os objetivos traçados.

É importante notar a dependência existente entre o nível intermediário, HAM, e o nível de Apresentação. Neste último, somente os recursos oferecidos pelo primeiro podem ser, de acordo com a conveniência, explorados.

A elaboração da interface com o usuário tem um papel fundamental no projeto de um sistema hipertexto. Devido ao alto grau de interatividade existente em sistemas dessa natureza, acredita-se até que alguns problemas inerentes a hipertexto possam ser contornados a partir de estudos na área de interface com o usuário ([24]). Deste modo, este tema voltará a ser discutido.

VII

Algumas Considerações sobre o Uso de Hipertexto

Não se pode negar a imensa flexibilidade existente no emprego de hipertexto. Porém, como fruto desta mesma flexibilidade, algumas situações indesejáveis podem ocorrer durante a navegação em hipertextos que se caracterizam, principalmente, pela agregação de um grande número de nós. Nestas situações, usuários que se encontram na busca de alguma informação se defrontam com questões do tipo:

"E agora, que caminho seguir?";

"Não era bem esta a informação que pensei estar acessando?";

"Como é mesmo que vim parar aqui?".

Estas questões caracterizam o chamado **paradigma da desorientação** em hipertextos que reflete situações em que, de modo geral, o navegante perde a noção da posição em que se encontra na rede de informações ou se vê incapacitado de acessar uma informação que ele sabe que existe (ou imagina que exista) no hipertexto.

A desorientação é um fenômeno que, além de ocorrer em hipertextos com um grande número de nós, é também verificada naqueles que apresentam várias opções de caminho durante a navegação ou que sofrem constantes alterações na estrutura ou que apresentam alguma deficiência na elaboração dessa mesma estrutura. Os dois últimos casos refletem a relação existente entre o processo de autoria e a desorientação, pois deve ser preocupação do autor a elaboração de estruturas compatíveis com as necessidades de exploração dos usuários.

É interessante notar que há também o risco de desorientação durante, por exemplo, a leitura de um livro, ou seja, o leitor pode perder a noção do **contexto** de leitura por distração ou por alguma interrupção súbita. Neste caso, porém, ele pode retornar ao início do capítulo ou ainda examinar o índice do livro para se situar novamente. Além disso, livros apresentam uma série de **ítems de orientação** que amenizam o risco de desorientação (Figura 15).

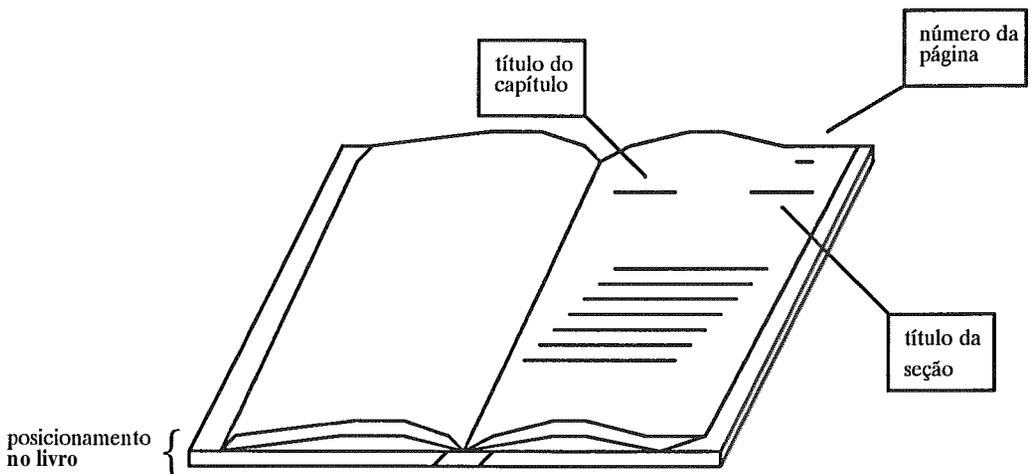


Figura 15 - Ítems de orientação

Com o intuito de evitar que a desorientação ocorra ou mesmo de auxiliar navegantes que já se encontram perdidos, muitos sistemas hipertexto oferecem recursos auxiliares ao processo de navegação que, em alguns casos, se baseiam em mecanismos já tradicionais* e, em outros, são específicos do meio computacional. Alguns exemplos são:

- . o conceito de **teia** introduzido pelo sistema Intermedia (veja subseção Intermedia, página 9) que restringe, em função do contexto, elos e nós disponíveis ao navegante;
- . os diagramas estruturais (**browser**) apresentados pelo sistema Notecards (veja subseção Notecards, página 10) que permitem exames na estrutura do hipertexto;
- . o mecanismo de **histórico** oferecido pelo sistema Hiperties que possibilita, durante a navegação, o exame da sequência de nós percorridos até um determinado instante.

*o emprego de índices, por exemplo, que é comum no meio literário, é utilizado por alguns sistemas hipertexto.

Em alguns casos específicos, o risco de desorientação pode ser tolerável. Basta imaginar um sistema hipertexto que seja utilizado, por exemplo, por visitantes de um museu em busca de informações sobre as obras em exibição. Nesta situação, os navegantes devem ser incentivados a fazerem incursões exploratórias, sem terem que se preocupar em manter controle sobre contextos percorridos.

A especificação dos recursos de auxílio que um sistema deve oferecer dependerá dos objetivos que se deseja alcançar com o mesmo e também da caracterização dos usuários que o operarão.

O Projeto

Introdução

Na fase anterior deste trabalho, discutiu-se Hipertexto com o objetivo de caracterizá-lo. Através da identificação de potencialidades e de problemas provenientes da exploração deste conceito, foi dado o primeiro passo para a sua efetiva utilização em meio computacional.

Nesta fase, a de Projeto, pretende-se formalizar um modelo que, atendendo a requisitos funcionais estabelecidos, viabilize o desenvolvimento de um sistema hipertexto. Para isso, três etapas serão cumpridas: o Levantamento de Requisitos Funcionais, o estudo de Aspectos da Interface com o Usuário e a Proposta de um Modelo. Todas estas serão futuramente detalhadas.

A Implementação, fase posterior à de Projeto, será, por último, descrita.

- . UNIX e OPEN LOOK são marcas registradas da AT&T.
- . OpenWindows, SunOs e XView são marcas registradas da Sun Microsystems.

I - O Levantamento de Requisitos Funcionais

Introdução

O Levantamento de Requisitos Funcionais deste projeto foi realizado a partir da caracterização de sistemas hipertexto existentes e do desenvolvimento de um protótipo, onde se verificou a viabilidade e a conveniência da incorporação, no sistema a ser desenvolvido, de algumas características.

Um resumo da etapa de caracterização será feito em duas seções: Tipos de Informação em Sistemas Hipertexto e Recursos do Processo de Navegação. Por último, serão relacionados os requisitos funcionais que deverão estar presentes no sistema em desenvolvimento.

I.1 – Tipos de Informação em Sistemas Hipertexto

A possibilidade de tratamento de informações não-textuais é extremamente enriquecedora para um sistema hipertexto*. Gráficos, sons e imagens, quando presentes, complementam o poder de expressão obtido pela utilização exclusiva de texto.

Um tipo de informação que deve ser destacado é aquele proveniente da incorporação de certos programas (simuladores, animadores, etc) ao hipertexto. Pode-se dizer que este tipo explora mais intensamente aspectos intuitivos da compreensão ao apresentar informações de maneira dinâmica, que é reconhecidamente um modo mais poderoso de se transmitir idéias.

Duas questões surgem relacionadas ao tratamento de tipos de informação em sistemas hipertexto: a extensibilidade de tipos e a compatibilidade entre padrões de armazenamento.

A questão da extensibilidade de tipos refere-se à viabilidade de se exibir algum tipo de informação que não seja previamente suportado pelo sistema. A possibilidade de se incorporar programas ao hipertexto pode ser uma maneira de se tratar esta questão, pois a apresentação de tais tipos poderia ser feita pelos próprios programas. Por exemplo, a apresentação de informação sonora poderia ser realizada por um programa.

*1 o termo hipermídia é geralmente utilizado quando o hipertexto é composto também por informações não-textuais. Neste trabalho, será empregado exclusivamente o termo hipertexto.

A existência de vários padrões de armazenamento de informação é, por outro lado, o cerne da questão da compatibilidade. É interessante que exista algum mecanismo que permita o aproveitamento de informações geradas em padrões alienígenas. A utilização de filtros como "canais de entrada" parece ser uma alternativa flexível (Figura 16).

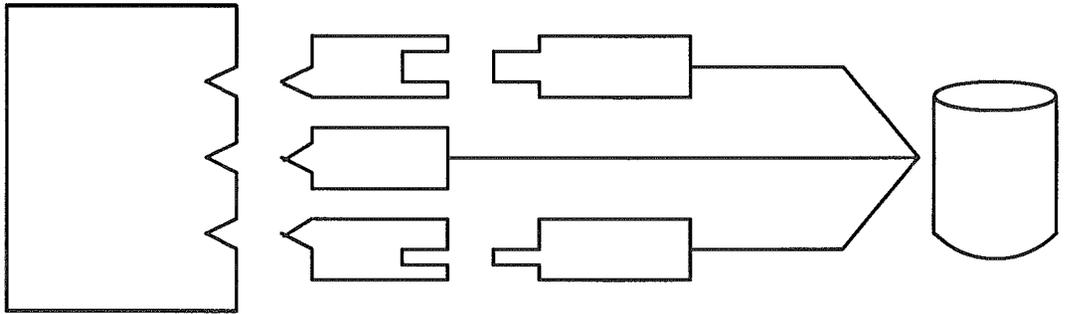


Figura 16 - Filtros como "canais de entrada"

É interessante notar que filtros podem ser utilizados, não somente na conversão entre padrões de armazenamento, como também para a geração em si de uma informação a ser apresentada.

A escolha dos tipos de informação que serão suportados pelo sistema é um ponto fundamental no projeto de um sistema hipertexto e será baseada principalmente nas características do ambiente de operação do sistema.

I.2 - Recursos do Processo de Navegação

Recursos do Processo de Navegação são aqueles que definem o processo, auxiliam na solução e na prevenção de problemas que este apresenta ou ainda enriquecem, de alguma maneira, o próprio ambiente de consulta. Por isso, a seleção daqueles que serão oferecidos representa um outro ponto importante no projeto de um sistema hipertexto.

Após a coletânea de recursos que são usualmente utilizados em alguns sistemas disponíveis atualmente foi feita uma pré-seleção (Figura 17) em que se considerou aqueles que seriam relevantes para o adequado uso do sistema no seu futuro ambiente de operação. Este último se caracteriza pela realização de trabalhos de pesquisa sobre assuntos intimamente relacionados, e também pela não utilização de ferramentas automatizadas tanto para a coordenação de tarefas, como para o intercâmbio entre pesquisadores.

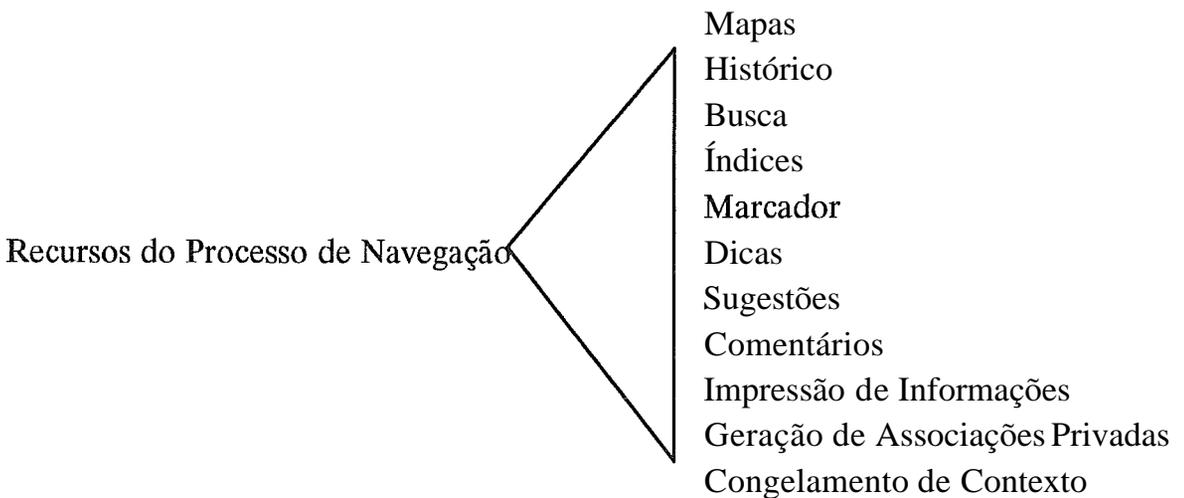


Figura 17

Com o intuito de determinar os recursos que realmente deverão ser colocados a disposição do navegador, a seguinte análise foi realizada:

Mapas

Mapas são representações gráficas dos nós e de suas associações. Podem ser globais ou locais quando representam, respectivamente, o universo ou um subconjunto das informações existentes (Figura 18). No segundo caso, um nó é sempre considerado como ponto de referência.

A função primordial de um mapa é obviamente a de localização. Nas situações em que o navegante perde a noção do contexto, ele pode requisitá-lo para se situar novamente. Alguns sistemas oferecem a possibilidade de seleção, no próprio mapa, de um próximo nó a ser visitado.

Quando o número de nós existentes ultrapassa um determinado limite, o mapa deixa de cumprir sua função, pois perde a legibilidade. Além disso, o carácter exploratório da navegação em hipertextos pode ser demasiadamente comprometido pela simples presença deste recurso.

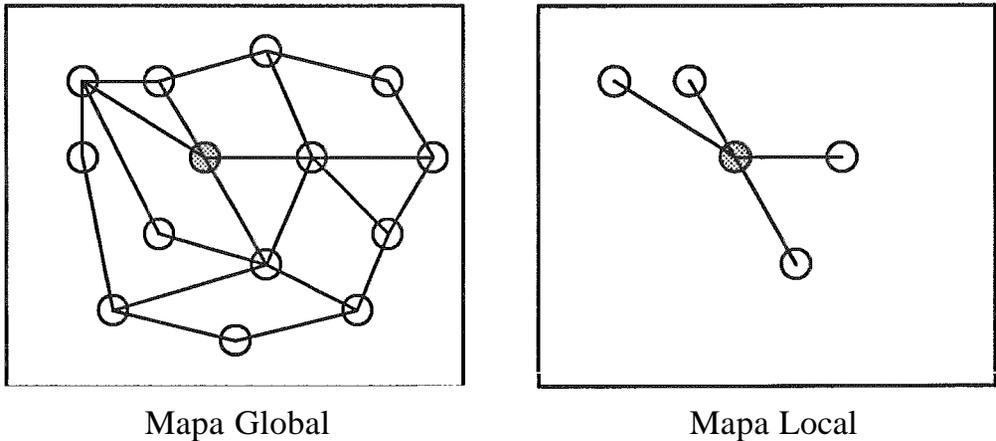


Figura 18

Não há nenhuma dúvida, porém, que a possibilidade de se ter uma visão panorâmica é fundamental no momento de construção de hipertextos, ou seja, na autoria. Em relação à consulta, fica claro que a utilidade deste recurso variará de situação em situação.

Histórico

Como o próprio nome diz, é um recurso que memoriza o caminho percorrido pelo navegante, para que este possa ser analisado em momento adequado.

De modo semelhante ao visto no uso de mapas, o Histórico pode ser requisitado quando o navegante perde a noção do contexto. Desta maneira, ele pode restabelecê-lo ao lembrar quais foram os nós percorridos até aquele instante.

Mais uma vez, a seleção de um nó pode ser feita através de indicação no próprio Histórico. Algumas outras facilidades, como por exemplo a Geração de Associações Privadas (discutido em breve) via Histórico, podem também estar presentes.

Busca

Busca pode ser considerado um mecanismo que possibilite o acesso direto a uma informação. Busca por palavras chaves é um dos exemplos de sua implementação.

A utilidade maior e, talvez, conveniência na exploração deste recurso se dá quando uma informação anteriormente acessada em um processo de navegação, deve ser novamente capturada. A existência de mecanismos de busca muito sofisticados poderia comprometer o "espírito aventureiro" dos navegantes.

Índices

Assim como aqueles encontrados em livros, índices organizam a informação segundo algum critério. Índices de títulos (de documentos) são os mais comuns.

Com emprego semelhante aos mecanismos de busca, índices podem ser utilizados para acesso direto a alguma informação do hipertexto.

Marcador

Marcador é um mecanismo que permite a marcação de nós estratégicos durante a navegação, permitindo um retorno direto em um momento qualquer (Figura 19). É baseado no hábito que alguns leitores têm, de marcarem as páginas interessantes de um livro durante a leitura.

É um recurso que pode ser útil para ajudar a solucionar o problema de desorientação em hipertextos.

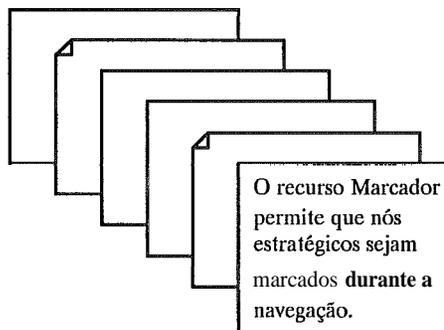


Figura 19

Dicas

Dicas servem para ajudar os navegantes na escolha dos caminhos. Quando um novo nó é selecionado, surge uma dica que oferece alguma informação complementar, além daquela dada pela própria chave de navegação, a respeito do conteúdo deste novo nó. Algumas incursões desnecessárias podem ser evitadas.

Sugestões

Nos ambientes em que são frequentes as navegações exploratórias, o navegante pode optar por seguir caminhos previamente definidos pelo autor. O hipertexto seria percorrido "tutorialmente".

Comentários

Podem ser comparados às anotações que algumas pessoas fazem na margem de livros. Existem dois tipos: os públicos e os privados. Os comentários públicos podem ser examinados pelos autores e por outros navegadores; os privados, só por seus autores.

Comentários aprimoram o ambiente de consulta, promovem a integração do grupo de navegadores e são extremamente úteis para os autores que podem revisar seus hipertextos utilizando sugestões dadas.

Impressão de Informações Apresentadas

É um recurso muito Útil, pois complementa o trabalho de consulta.

Geração de Associações Privadas ([25])

Com a disponibilidade deste recurso, os navegantes podem gerar suas próprias associações em função de suas necessidades. Os autores terão, por sua vez, mais uma maneira de reavaliarem seus trabalhos, pois o exame de novas propostas de associações pode indicar alguma falha na elaboração da estrutura do hipertexto, ocorrida durante o processo de autoria.

A Geração de Associações Privadas é um processo de síntese, sendo, portanto, de natureza complementar ao de análise que a navegação em hipertextos sugere. A combinação desses dois fatores pode ser extremamente conveniente em algumas situações.

Congelamento de Contexto

Recurso que se caracteriza pela memorização do contexto* de uma sessão, caso o navegante tenha que interrompê-la. Deste modo, ele pode continuar a navegação do mesmo ponto e na mesma situação em que parou.

*em sistemas hipertexto em que nós são apresentados através de um sistema de janelas, o contexto de uma sessão representaria, entre outras coisas, as janelas que estão abertas e suas posições na tela.

I.3 – Os Requisitos Funcionais

Em função do ambiente de operação do sistema e da avaliação funcional realizada pelo desenvolvimento de um protótipo, os seguintes requisitos funcionais foram estabelecidos:

- . tipos de informação que o sistema deverá suportar:
 - . textos (sequência de caracteres);
 - . gráficos (desenho ou figura vetorial);
 - . imagens (sequência de elementos que podem ser associados a uma área da tela do computador para se obter alterações cromáticas);
 - . programas (sequência de instruções que podem ser executadas no computador).

Em uma primeira versão do sistema, imagens, gráficos e programas, quando presentes, serão nós terminais do hipertexto;

- . recursos auxiliares ao Processo de Navegação que o sistema deverá oferecer:
 - . Histórico, Marcador, Impressão, Comentários, Mapas, Dicas e Sugestões;
- . requisitos adicionais:
 - . deve haver algum mecanismo que viabilize a apresentação de informações que foram geradas em formatos não-compatíveis com o do sistema;
 - . deve existir algum mecanismo que possibilite a apresentação de tipos de informação não previamente suportados.

A partir destes requisitos, pretende-se desenvolver um sistema que possa cumprir o objetivo principal de integrar o ambiente de pesquisa no qual irá operar.

II - Aspectos da Interface com o Usuário

Introdução

O projeto da interface com o usuário de um sistema é uma tarefa complexa, pois uma série de fatores devem ser considerados. Profissionais de diversas áreas têm se dedicado a este tema com o objetivo primordial de criarem mecanismos que possibilitem o desenvolvimento de sistemas realmente "amigáveis".

No caso específico do projeto de sistemas hipertexto, uma grande ênfase deve ser dada na elaboração da interface com o usuário, visto que estes sistemas são inerentemente interativos. Acredita-se até que alguns problemas existentes em sistemas desta natureza (veja referência 24 na Bibliografia, página 77) possam ser resolvidos a nível de suas interfaces com o usuário.

Nos parágrafos seguintes, serão discutidas algumas questões gerais sobre o tema interface com o usuário, visando uma análise subsequente, relacionada mais especificamente a hipertexto. Com isso, pretende-se estabelecer um critério a ser considerado na especificação de uma interface que atenda às necessidades de interação do sistema aqui projetado.

II.1 - A Questão da Usabilidade

De modo geral, a interface com o usuário de um sistema permite que uma funcionalidade pré-estabelecida seja colocada a disposição dos usuários. Obviamente, o objetivo principal é criar um ambiente no qual seja necessário um esforço mínimo para se explorar as funções disponíveis, ou seja, um ambiente amigável.

Durante o processo de desenvolvimento da interface de um sistema, o projetista propõe alternativas que podem ser avaliadas pelos chamados **testes de usabilidade**. Nestes testes, pretende-se medir a qualidade no uso do sistema e isto é feito através da avaliação dos seguintes parâmetros:

. tempo de aprendizado: refere-se ao tempo que os usuários do ambiente levam para aprender a executar funções suportadas pelo sistema;

- . eficiência no uso: refere-se à viabilidade dos usuários atingirem os objetivos desejados quando utilizam o sistema;
- . taxa de erros: é um parâmetro crítico. Refere-se ao número e aos tipos de erros que os usuários cometem enquanto estão usando o sistema;
- . retenção: refere-se à facilidade que usuários têm de reutilizarem o sistema após um período de afastamento;
- . satisfação no uso: é um parâmetro muito subjetivo. Tenta-se avaliar a satisfação que os usuários têm ao utilizarem o sistema. Este parâmetro está intimamente relacionado aos demais.

Como se vê, os parâmetros para medição de usabilidade formam um critério subjetivo para a avaliação de interfaces com o usuário. A grande interdependência entre estes parâmetros, juntamente com questões como a de diferenças pessoais entre os usuários de um ambiente (experiências, habilidades perceptuais e cognitivas) fazem com que o projeto de uma interface seja uma tarefa elaborada. Um ponto claro é que um intensivo processo de avaliação e refinamento, com o desenvolvimento, se possível, de protótipos deva ser executado. Só assim, o projetista da interface terá êxito em sua missão que é de fundamental importância para o sucesso operacional de um sistema.

Nos próximos parágrafos, será abordada a questão de interface com o usuário relacionada especificamente a sistemas hipertexto.

II.2 – Hipertexto e sua Funcionalidade Básica

A grande importância na elaboração da interface com o usuário em um projeto de sistema hipertexto é derivada da alta dinamicidade do processo de navegação. Este processo pode ser sumariamente descrito pela sucessiva apresentação de informações associadas. Deste modo, duas questões devem ser fundamentalmente consideradas a nível de interface com o usuário: como apresentar as informações e como representar as associações. A seguir, será feita uma breve discussão sobre essas questões.

A Apresentação de Informações

A consideração mais importante a respeito de como se apresentar informações relaciona-se à conveniência de se utilizar janelas de exibição, pois poderia-se optar pela utilização, a cada nó percorrido, da área inteira da tela para a apresentação das informações associadas.

No caso do uso de janelas, um controle cuidadoso na disposição das mesmas poderia colaborar para amenizar o problema de perda de contexto, já que o "percurso" percorrido pelo navegante estaria sempre estampado na tela (Figura 20); por outro lado, o processo de navegação poderia ser prejudicado para usuários menos experientes, pois o controle de várias janelas seria uma preocupação latente. Como se vê, a caracterização dos usuários do sistema é um fator decisivo neste caso.



Figura 20

Neste projeto, em função do ambiente de operação do sistema, onde já é comum a utilização de sistemas que operam com janelas, e da experiência computacional dos futuros usuários, optou-se por apresentar as informações em janelas. Além disso, esta opção parece melhor condizer com a diretriz básica utilizada no desenvolvimento do modelo deste sistema no qual cada nó do hipertexto será representado por um processo do Sistema Operacional (veja seção III.1 – O Modelo, página 44). É razoável pensar que cada um destes processos tenha uma janela de exibição própria.

Vale lembrar que o sistema deverá suportar a apresentação de diversos tipos de informação. Além disso, como característica adicional, em um mesmo nó, poderão estar vinculados mais de um **bloco de informação** que podem, portanto, ser de tipos diferentes. Deste modo, cada janela de exibição deverá poder ser dividida em subjanelas e estas sim serão responsáveis pela exibição de textos, imagens ou de programas (Figura 21).

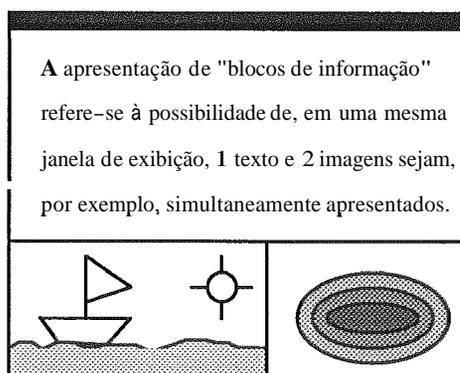


Figura 21 - A apresentação de blocos de informação

A Representação de Chaves de Navegação

Chaves de navegação são indicações nas informações apresentadas que, ao serem ativadas, originam a apresentação de outros nós do hipertexto, ou seja, implicam na criação de novas janelas de exibição. Suas ativações devem ser preferencialmente feitas através de um mecanismo simples, como por exemplo, com interação através de cursor/ratinho.

Existem diversas maneiras de se destacar chaves de navegação*¹: com a utilização de fontes diferentes, realçando-as com cores contrastantes, delimitando-as com sinais específicos etc (Figura 22). Neste sistema, por existirem várias restrições a nível do ambiente de implementação*² que impossibilitam a escolha de qualquer outra alternativa, foi decidido que as áreas de texto que correspondam a chaves de navegação serão delimitadas com sinais específicos.

*¹ Na primeira versão do sistema, só estão sendo consideradas chaves de navegação em informações textuais (veja seção 1.3 - Os Requisitos Funcionais, página 37).

*² Restrições do tipo: falta de flexibilidade na manipulação de fontes; impossibilidade de utilização de cores na impressão de caracteres etc.

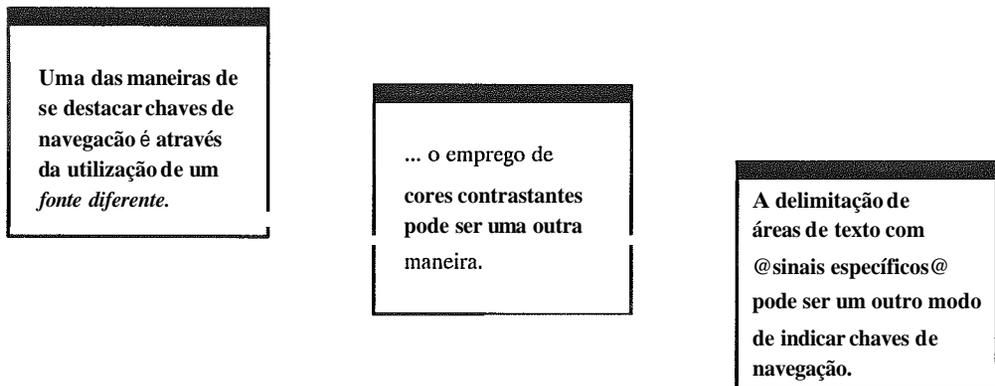


Figura 22 – Destacando chaves de navegação

II.3 – Hipertexto e Usabilidade

No caso específico de sistemas hipertexto, um dos maiores problemas que existe em relação à usabilidade é o risco de desorientação durante o processo de navegação. Desde modo, levando-se em consideração os aspectos anteriormente analisados, os parâmetros para avaliação de usabilidade devem ser considerados da seguinte maneira:

- . tempo de aprendizado: deve ser o mais breve possível, ou seja, os comandos básicos disponíveis devem poder ser facilmente acessáveis por usuários que utilizem o sistema, por exemplo, pela primeira vez;
- . eficiência: partindo do pressuposto que navegadores estão em busca de alguma informação, eles devem poder rapidamente encontrá-las, ou então descobrir que elas não estão presentes no hipertexto;
- . taxa de erros: as funções do sistema devem estar disponíveis de maneira clara, evitando assim o inconveniente de solicitações serem feitas pelos usuários e não serem atendidas de modo adequado;
- . retenção: depois de um período sem usarem o sistema, os navegadores não devem ter problemas para lembrarem, por exemplo, como navegar novamente por hipertextos;

. satisfação no uso: a satisfação no uso depende, entre outras coisas, da utilidade do sistema no ambiente de operação, pois este é um fator de estímulo para o seu próprio uso. É um parâmetro que, mesmo subjetivo, pode ser avaliado pela coleta de opiniões dos usuários.

Um ponto importante a se considerar é que alguns problemas de usabilidade podem ser originados por falhas na estruturação do hipertexto, ou seja, a navegação em hipertextos "mal elaborados" pode comprometer a orientação dos navegantes, acarretando constantes inconvenientes como perdas de contexto.

Deste modo, a autoria de hipertextos é um processo de fundamental importância quando se considera, de modo mais abrangente, a questão de usabilidade em sistemas hipertextos.

. . .

Nesta seção, foram discutidas algumas questões relacionadas à interface com o usuário de hipertextos que, associadas aos princípios básicos de simplicidade, consistência e eficiência, pudessem facilitar a especificação da interface do sistema aqui projetado. Como já fora mencionado, esta tarefa é de fundamental importância devido ao carácter predominantemente interativo de sistemas hipertexto.

III - A Proposta de um Modelo

Introdução

Nesta etapa do projeto, a seguinte questão é levantada: Como estruturar um sistema hipertexto que ofereça uma série de recursos pré-estabelecidos?

Para respondê-la, um modelo será inicialmente apresentado e posteriormente detalhado, que serviu de base à implementação descrita no capítulo IV.

III.1 - O Modelo

Dentre as muitas definições existentes de Hipertexto, uma que parece razoável, nesta situação, é a seguinte:

"Hipertexto é uma rede de informações, associado a um mecanismo que possibilite percorrer livremente esta rede, acessando as informações".

Além disso, vale lembrar que o percurso da rede deve ser acompanhado e registrado de maneira tal que viabilize a incorporação dos recursos auxiliares ao processo de navegação, presentes na especificação de requisitos funcionais do sistema (veja seção 1.3 – Os Requisitos Funcionais, página 37).

Na Figura 23, é apresentado o esquema de um modelo que tem como fundamento básico a exploração, a nível de Sistema Operacional, do conceito de **processo*** e de todas as facilidades associadas (mecanismos de comunicação entre processos (IPC), chamadas ao Sistema, etc). Com exceção ao ARQUIVO HIPERTEXTO, todos os demais componentes do modelo são processos.

O ARQUIVO HIPERTEXTO tem o papel fundamental de armazenar, não somente a estrutura do hipertexto a ser percorrido, como também os nomes dos arquivos que contém as informações (textos, imagens, programas etc) a serem apresentadas. Deste modo, no ARQUIVO HIPERTEXTO, estão determinadas todas as informações associadas a cada nó do hipertexto.

O GERENCIADOR DE ATIVAÇÃO é o processo responsável pela ativação de todos os PROCESSOS APRESENTADORES. Sendo assim, tanto o início, como o cancelamento de um processo de navegação se dão por seu intermédio.

* processo é uma instância de um programa que está sendo executada pelo Sistema Operacional.

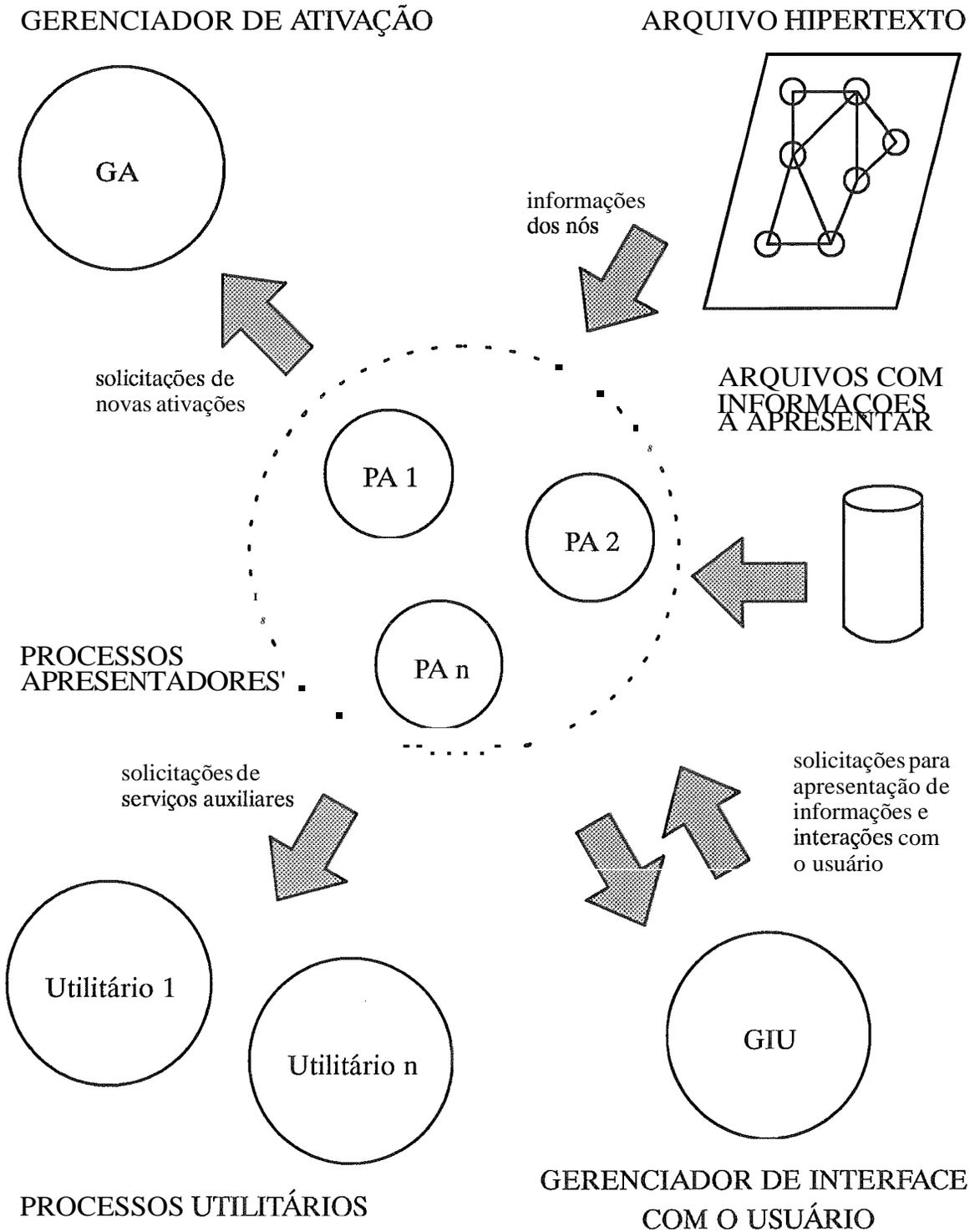


Figura 23 – O Modelo

Os PROCESSOS APRESENTADORES representam os nós ativos do hipertexto, pois apresentam, em janelas individuais do dispositivo de exibição, informações desses nós. No momento inicial de suas ativações, eles consultam o ARQUIVO HIPERTEXTO para acessarem as informações associadas ao nó que representam.

Além de representarem nós do hipertexto, os PROCESSOS APRESENTADORES têm um papel fundamental na dinâmica da navegação, armazenando associações e solicitando ao GERENCIADOR DE ATIVAÇÃO, quando for o caso, a ativação de outros processos apresentadores, ou seja, a apresentação de outros nós.

Os PROCESSOS UTILITÁRIOS são processos que colocam a disposição do usuário recursos auxiliares ao processo de navegação. Deste modo, de acordo com os requisitos funcionais deste projeto, os recursos Histórico, **Marcador**, Impressão, Comentários, Mapas, Dicas e Sugestões serão viabilizados através do desenvolvimento de seus respectivos utilitários que são programas independentes que, ao serem ativados, serão responsáveis pela disponibilidade dos recursos associados.

O GERENCIADOR DE INTERFACE COM O USUÁRIO é o processo reponsável pelo fornecimento de recursos que viabilizam a apresentação propriamente dita de informações e a interação usuário/sistema. Tem, portanto, um papel fundamental, visto que sistemas hipertexto são inerentemente interativos e o modo como as informações são apresentadas é um fator de grande importância.

. . .

A idéia de se explorar como fundamento básico, no projeto de um sistema hipertexto, o conceito de processo tem origem no requisito que determina a possibilidade de se agregar ao hipertexto informações de natureza dinâmica (programas, animações, etc).

Diante da constatação que questões como a extensibilidade de tipos e a compatibilidade entre padrões de armazenamento poderiam ser mais facilmente tratadas, optou-se por estender essa idéia o mais amplamente possível. Sendo assim, a apresentação de qualquer informação seria feita também por processos independentes.

De maneira análoga, visando uma possível extensão da funcionalidade do sistema em relação a recursos de navegação, o conceito de "PROCESSOS UTILITÁRIOS" foi elaborado. Deste modo, novos recursos podem ser agregados por intermédio do desenvolvimento de seus respectivos utilitários. Vale notar que a incorporação de um novo utilitário ao sistema se dará através do mecanismo **programa ponte** (veja a seguir em A Dinâmica do Modelo) e **não** implicará em qualquer alteração de código já desenvolvido para os Processos Apresentadores.

Em uma análise preliminar, não parece haver qualquer empecílio que impossibilite ou, até mesmo, desestímule tal exploração.

III.2 – A Dinâmica do Modelo

Durante o processo de navegação em hipertextos, nós/processos são visitados/ativados e recursos de auxílio são solicitados. Tudo isto implica em uma série de procedimentos internos ao modelo que são executados em função de estímulos externos (solicitações do navegador). A Dinâmica do Modelo reflete o relacionamento entre os componentes do modelo, subjacente a esses procedimentos internos.

Inicialmente, antes que um processo de navegação se inicie, os Processos Utilitários e o Gerenciador de Ativação devem ser instalados. No momento de iniciar uma navegação, o usuário deve informar ao Gerenciador de Ativação o hipertexto que será percorrido. Este se encarrega, então, de ativar o processo apresentador correspondente ao nó inicial deste hipertexto (esta informação está também registrada no Arquivo Hipertexto).

Os processos apresentadores, ao serem ativados, são informados sobre que nó representam. Deste modo, eles consultam o Arquivo Hipertexto em busca das informações associadas a este nó, ou seja: um rótulo, o nome de um arquivo que especifica o "layout" de sua janela de exibição, o nome de um arquivo que contém uma série de chamadas de programas que serão executadas neste momento de inicialização, o nome de um arquivo que contém uma série de chamadas de programas que serão executados no momento de sua finalização e os dados necessários para a apresentação de todas as informações vinculadas. Após solicitação ao Gerenciador de Interface com o Usuário dos recursos necessários à apresentação propriamente dita da informação (por exemplo: uma subjanela para exibição de informações textuais e duas subjanelas para exibição de imagens), os processos apresentadores cumprem o seu papel funcional, apresentando as informações vinculadas, e permanecem aguardando requisições do navegador que, neste momento, Já pode ativar uma chave de navegação qualquer.

É importante destacar que a execução de chamadas de programas, na inicialização e na finalização de um processo apresentador, constitui um mecanismo importante para a efetiva implementação da idéia de **Processos Utilitários**. É através deste mecanismo, juntamente com a possibilidade de execução de chamadas durante a "vida útil" de um processo apresentador*¹, que os Processos Utilitários obtêm as informações necessárias ao cumprimento de suas funções. Programas executados na inicialização ou na finalização de um processo apresentador com este objetivo específico são chamados programas **ponte***² (Figura 24).

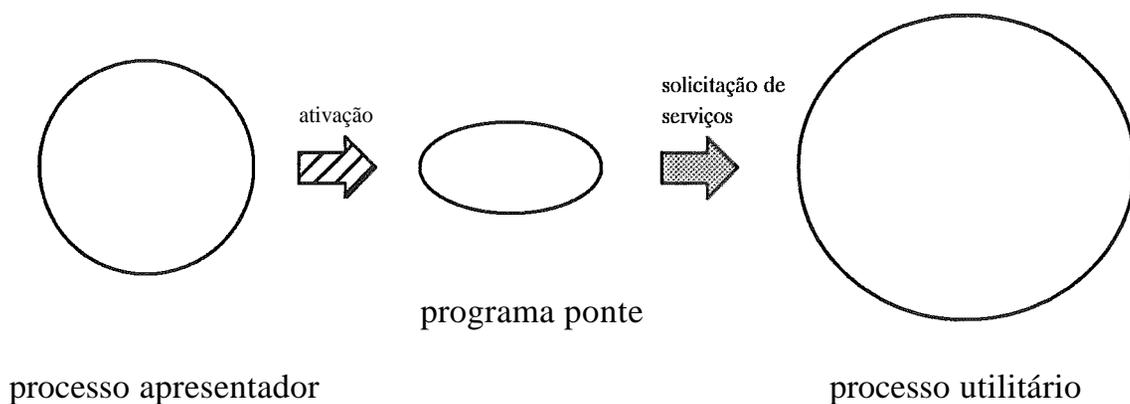


Figura 24 - O funcionamento dos programas ponte

É interessante notar que a comunicação entre os processos apresentadores e os Processos Utilitários poderia ser feita diretamente através da incorporação de bibliotecas de comunicação na estrutura dos próprios processos apresentadores. Neste caso, porém, caso um novo utilitário fosse incorporado ao sistema, a sua respectiva biblioteca teria que ser também incorporada na estrutura dos processos apresentadores, o que implicaria em alterações de código já desenvolvido. Por esta razão optou-se pelo mecanismo **programa ponte**.

Vale ressaltar que este mecanismo descrito por último pode ser utilizado também para a extensão da funcionalidade do sistema, sendo, por isso, de grande importância neste projeto.

*1 estas chamadas serão executadas quando os usuários solicitarem algum recurso de navegação através da interface com o(s) proc.(s) apr.(s). Elas estarão descritas em um arquivo que será acessado pelo(s) proc.(s) apr.(s) no momento de sua(s) ativação(s).

*2 o nome **programa ponte** remete à idéia de uma "ponte" de comunicação entre os processos apresentadores e os Processos Utilitários.

A seguir serão detalhados os componentes básicos do modelo, que são: o Arquivo Hipertexto, os processos apresentadores, o Gerenciador de Ativação e o Gerenciador de Interface com o Usuário.

III.3 – O Detalhamento do Modelo

O Arquivo Hipertexto, os processos apresentadores, o Gerenciador de Ativação e o Gerenciador de Interface com o Usuário são os componentes básicos do modelo apresentado anteriormente, base para o desenvolvimento do sistema. Nos parágrafos seguintes, todos eles serão detalhados, tendo em vista a fase de Implementação que virá a seguir.

O Arquivo Hipertexto

O processo de navegação em hipertextos consiste na apresentação de informações associadas. Neste projeto, optou-se por armazenar essas informações em arquivos isolados, mantendo as associações juntamente com as referências a esses arquivos em um único arquivo, o Arquivo Hipertexto.

Os processos apresentadores consultam o Arquivo Hipertexto, no momento inicial de suas ativações, em busca das seguintes informações associadas ao nó que representam: um rótulo que será exibido quando o nó for apresentado, o nome de um arquivo que especifica o "layout" da janela onde serão exibidas as informações, o nome de um arquivo que contém chamadas de programas que serão executadas durante a sua inicialização, o nome de um arquivo que contém chamadas de programas que serão executadas antes de sua finalização e os dados necessários para a apresentação de todas as informações vinculadas, que são: os nomes dos arquivos que armazenam as informações, os nomes dos respectivos programas filtros e as chaves de navegação. Para isso, o Arquivo Hipertexto possui o seguinte formato*:

*as chaves ({}) que aparecem na descrição do arquivo indicam, aos pares, repetição de alguma informação.

```

nNós
{idNó posiçãoNoArquivo}nNós
chamadaDoPrimeiroProcessoApresentador
nomeDoProgramaFiltroPadrãoParaTextos
nomeDoProgramaFiltroPadrãoParaImagens
nomeDoArquivoDeInicializaçãoPadrão
nomeDoArquivoDeFinalizaçãoPadrão
nomeDoArquivoComLayoutPadrão
{
  idNó
  rótulo
  nArquivosComInformação
  {
    nomeDoProgramaFiltroEspecífico
    tipoDaInformação
    parâmetrosDoProgramaFiltro
    nChavesDeNavegação
    {
      regiãoDaInformação
      chamadaDeProcessoApresentador
    }nChavesDeNavegação
  }nArquivosComInformação
  nomeDoArquivoDeInicializaçãoEspecífico
  nomeDoArquivoDeFinalizaçãoEspecífico
  nomeDoArquivoComLayoutEspecífico
}nNós

```

Logo no início do Arquivo Hipertexto, existe uma tabela que serve para facilitar o trabalho de busca de informações que os processos apresentadores executam quando são ativados. Os seus elementos são:

. nNós é um valor inteiro que representa o número de nós que estão descritos dentro do arquivo;

idNó é uma sequência de caracteres que representa o identificador de um nó;

. posiçãoNoArquivo é um valor inteiro que representa a localização, dentro do arquivo, das informações vinculadas a um nó, sendo expressa pelo número de bytes que separam o fim da tabela e o início da descrição das informações.

Os próximos elementos são os seguintes:

. chamadaDoPrimeiroProcessoApresentador*¹ é uma sequência de caracteres que representa a chamada do processo apresentador que será ativado no momento inicial da navegação do hipertexto;

. nomeDoProgramaFiltroPadrãoParaTexto*² é uma sequência de caracteres que representa o nome de um programa filtro que pode ser usado para o tratamento de informações textuais vinculadas a nós do hipertexto. Caso não exista um programa filtro padrão para texto, uma sequência vazia deverá substituir o nome do programa;

. nomeDoProgramaFiltroPadrãoParaImagens é uma sequência de caracteres que representa o nome de um programa filtro que pode ser usado para o tratamento de imagens vinculadas a nós do hipertexto. Caso não exista um programa filtro padrão para imagens, uma sequência vazia deverá substituir o nome do programa;

. nomeDoArquivoDeInicializaçãoPadrão*³ é uma sequência de caracteres que representa o nome de um arquivo que contém uma série de chamadas de programas que podem ser ativadas no momento inicial da ativação de um processo apresentador. Caso este arquivo não seja utilizado, uma sequência vazia deverá substituir o nome do arquivo;

. nomeDoArquivoDeFinalizaçãoPadrão*³ é uma sequência de caracteres que representa o nome de um arquivo que contém uma série de chamadas de programas que podem ser ativadas no momento final da ativação de um processo apresentador. Caso este arquivo não seja utilizado, uma sequência vazia deverá substituir o nome do arquivo;

*1 chamadas de processos apresentadores serão descritas na seção de detalhamento destes mesmos processos.

*2 o funcionamento de programas filtro será detalhado mais adiante.

*3 o funcionamento de arquivos de inicialização e de finalização será detalhado mais adiante.

nomeDoArquivoComLayoutPadrão* é uma sequência de caracteres que representa o nome de um arquivo que contém uma descrição de janela de exibição que pode ser utilizada na apresentação das informações vinculadas a um nó do hipertexto. Caso não exista um arquivo com "layout" padrão, uma sequência vazia deverá substituir o nome do arquivo;

A seguir, serão descritas as informações vinculadas especificamente a cada nó do hipertexto.

. idNó (já descrito anteriormente);

. rótulo é uma sequência de caracteres que representa um rótulo que será exibido durante a apresentação do nó. Caso não exista um rótulo, uma sequência vazia deverá substituí-lo;

. nArquivosComInformação é um valor inteiro que representa o número de blocos de informação (veja Figura 21, página 41) que serão apresentados;

Para cada bloco de informação, está associado um conjunto que é composto dos próximos seis elementos.

. nomeDoProgramaFiltroEspecífico é um conjunto de caracteres que representa o nome de um programa filtro que será usado para o tratamento da informação que está descrita em "parâmetrosDoProgramaFiltro" (veja a seguir). Caso não exista um programa filtro específico, uma sequência vazia deverá substituir o nome do programa;

tipoDaInformação é um valor inteiro, compreendido entre 0 e 3, que representa o tipo de informação que está sendo considerada neste bloco e que obedece à seguinte convenção:

"tipoDaInformação"	= 0	-> informação textual
	= 1	-> imagem
	= 2	-> programa que usará uma área interna da janela de exibição durante a sua execução
	= 3	-> programa que não usará uma área interna da janela de exibição durante a sua execução

* arquivos com "layout" serão detalhados mais adiante.

- . parâmetrosDoProgramaFiltro é uma sequência de caracteres que representa todas as informações que são passadas a um programa filtro para a geração da informação que será realmente apresentada;
- . nChavesDeNavegação é um número inteiro que representa o número de chaves de navegação que estão presentes neste bloco de informação;
- . regiãoDaInformação é uma sequência composta por dois valores inteiros que representam os limites de uma região da informação que é considerada chave de navegação;
- . chamadaDeProcessoApresentador é uma sequência de caracteres que representa a chamada do processo apresentador que será ativado, caso a região da informação correspondente seja apontada;

Por último, associados também a cada nó, existem três elementos que são descritos a seguir.

- . nomeDoArquivoDeInicializaçãoEspecífico é uma sequência de caracteres que representa o nome de um arquivo que contém uma série de chamadas de programas que serão ativadas no momento inicial da ativação do processo apresentador associado ao nó em questão. Caso não exista um arquivo de inicialização específico, uma sequência vazia deverá substituir o seu nome;
- . nomeDoArquivoDeFinalizaçãoEspecífico é uma sequência de caracteres que representa o nome de um arquivo que contém uma série de chamadas de programas que serão ativadas no momento final da ativação do processo apresentador associado ao nó em questão. Caso não exista um arquivo de finalização específico, uma sequência vazia deverá substituir o seu nome;
- . nomeDoArquivoComLayoutEspecífico é uma sequência de caracteres que representa o nome de um arquivo que contém a descrição da janela de exibição que será utilizada para a apresentação propriamente dita das informações vinculadas ao nó em questão. Caso não exista um arquivo com layout específico, uma sequência vazia deverá substituir o nome do arquivo.

Como complemento à descrição do Arquivo Hipertexto, os seguintes tópicos serão abordados:

- . O Funcionamento dos Programas Filtro;
- . Os Arquivos de Inicialização e Finalização;
- . O Arquivo com Layout.

O Funcionamento dos Programas Filtro

Programas Filtro são aqueles utilizados pelos processos apresentadores para converterem informações que estejam em formatos incompatíveis com o do sistema (veja Apêndice B do Projeto – Formatos Padrão do Sistema, página 72) para formatos compatíveis.

De modo geral, no momento em que um processo apresentador vai apresentar um bloco de informação, ele verifica se existe um programa filtro específico associado àquele bloco. Caso exista, ele utiliza este programa filtro, porém, caso não exista, ele verifica, a partir do tipo de informação do bloco, se foi especificado um programa filtro padrão. Este procedimento pode levar a duas situações distintas: uma em que não foi constatada a existência de um programa filtro e a outra em que foi. No primeiro caso, não haverá filtragem; no segundo, a filtragem se dará através da execução da seguinte linha de comando*:

```
programaFiltro parâmetrosDoProgramaFiltro > arquivoComInformaçãoFiltrada
```

Como se vê, a filtragem de informações resulta na criação de um arquivo temporário (**arquivoComInformaçãoFiltrada**), que será utilizado durante a exibição do bloco de informação em questão.

*esta linha de comando segue a sintaxe do interpretador de comandos do Sistema Operacional SunOs.

Os Arquivos de Inicialização e Finalização

Os Arquivos de Inicialização e Finalização são compostos de uma série de chamadas de programas que são executadas respectivamente no início e no final da ativação de um processo apresentador. Cada um desses arquivos pode ser descrito da seguinte maneira:

$$\{\text{chamadaDePrograma}\}_{n\text{Chamadas}}$$

onde:

- . chamadaDePrograma é uma sequência de caracteres que representa a chamada de um programa qualquer;
- . nChamadas é um valor inteiro que representa o número total de chamadas de programas existentes no arquivo.

Por último, vale lembrar que o mesmo critério adotado para a determinação do programa filtro a ser efetivamente utilizado no tratamento de uma informação associada a um nó (programa filtro "específico" x programa filtro "padrão"), será também utilizado aqui, para determinar o arquivo, de Inicialização ou de Finalização, de onde serão extraídas as chamadas de programa.

O Arquivo com Layout

O Arquivo com Layout possui a descrição da janela de exibição que, fornecida pelo Gerenciador de Interface com o Usuário, será utilizada para a apresentação propriamente dita das informações de um nó. Como em um mesmo nó podem estar vinculados mais de um bloco de informação, uma janela de exibição pode ser composta de várias subjanelas; uma para cada bloco.

A estrutura do Arquivo com Layout é a seguinte:

dimensõesDaÁreaDeExibição

nSubjanelas

{

tipoDeInformaçãoDaSubjanela

posiçãoDaSubjanela dimensõesDaSubjanela

}nSubjanelas

onde:

- . dimensõesDaÁreaDeExibição é um conjunto composto de dois valores inteiros que representam respectivamente a largura e a altura da janela de exibição;
- . nSubjanelas é um valor inteiro que representa o número de subjanelas que compõem a janela de exibição;

A cada subjanela, estão associados os seguintes elementos:

- . tipoDeInformaçãoDaSubjanela é um valor inteiro, compreendido entre 0 e 3, que representa o tipo de informação que será exibido na subjanela (veja a descrição de tipoDaInformação, página 52);
- . posiçãoDaSubjanela é um conjunto composto de dois valores inteiros que representam as coordenadas do canto superior esquerdo da subjanela, relativas ao interior da janela de exibição;
- . dimensõesDaSubjanela é o conjunto de dois valores inteiros que representam, nesta ordem, a largura e a altura da subjanela.

Mais uma vez, o critério para determinação do arquivo com layout de um nó será análogo ao utilizado para determinação de programas filtro e arquivos de inicialização e finalização ("específico x padrão").

. . .

O Gerenciador de Ativação

O Gerenciador de Ativação é o processo servidor responsável pela ativação de processos apresentadores. Existem duas situações distintas em que processos apresentadores são ativados: a primeira refere-se à apresentação do nó inicial de um hipertexto no primeiro momento da navegação. Neste caso, o Gerenciador de Ativação varre o arquivo hipertexto em questão, em busca da chamada do primeiro processo apresentador. A segunda, mais frequente, refere-se a solicitações que processos apresentadores fazem, durante a navegação, de ativações de outros processos apresentadores.

A interação entre navegante e Gerenciador de Ativação é realizada através de uma janela de controle que é criada no momento de sua ativação. Deste modo, o Gerenciador de Ativação também utiliza recursos providos pelo Gerenciador de Interface com o Usuário.

A interface com os processos apresentadores é, por sua vez, estabelecida por intermédio de uma biblioteca de comunicação que é incorporada à estrutura desses processos. Esta biblioteca será descrita a seguir.

. . .

Na implementação da biblioteca de comunicação entre processos apresentadores e o Gerenciador de Ativação, serão utilizadas as chamadas ao sistema (*system calls*) que operacionalizam mecanismos de IPC (*Interprocess Communication*) do sistema operacional UNIX. Por não estarem dentro do escopo deste projeto, detalhes de implementação serão omitidos.

função*: GA_processoApresentadorEncerrando(idGA,idPA)

utilização: esta função é utilizada pelos processos apresentadores, para avisarem ao Gerenciador de Ativação que estão encerrando;

parâmetros:

. idGA é um valor inteiro que representa o identificador, a nível do Sistema Operacional, do Gerenciador de Ativação;

. idPA é um valor inteiro que representa o identificador, a nível do Sistema Operacional, do processo apresentador que chama a função.

função: GA_ativaProcessoApresentador(idGA,
idPA,
nomeDoArquivoHipertexto,
idNó,
xDaJanela,ydaJanela)

utilização: esta função é utilizada pelos processos apresentadores, para solicitarem ao Gerenciador de Ativação a ativação de outros processos apresentadores;

*GA, prefixo do nome das funções, refere-se a Gerenciador de Ativação.

parâmetros:

. **idGA** e **idPA** são os mesmos descritos anteriormente;

nomeDoArquivoHipertexto, **idNó**, **xDaJanela** e **yDaJanela** são os parâmetros que compõe a chamada do processo apresentador a ser ativado (veja subseção Chamadas de Processos Apresentadores, página 59).

Vale lembrar que antes que qualquer processo de navegação se inicie, o Gerenciador de Ativação deve ser impreterivelmente instalado, quando permanece aguardando solicitações de ativação de processos apresentadores.

. . .

Os Processos Apresentadores

Os processos apresentadores têm o papel fundamental de apresentarem informações dos hipertextos, ou seja, representam, na verdade, nós destes hipertextos. Além disso, armazenam em suas estruturas internas os identificadores dos processos que serão executados, caso as chaves de navegação correspondentes, presentes nas informações apresentadas, sejam ativadas. Deste modo, os processos apresentadores têm também função estrutural, pois armazenam associações do hipertexto.

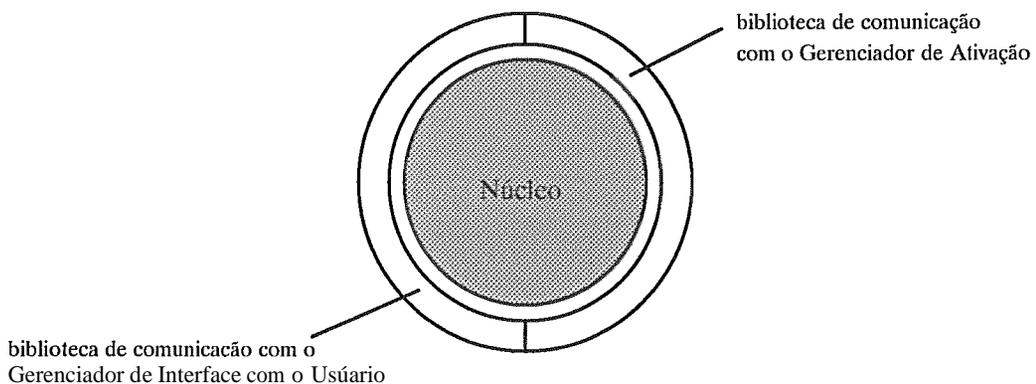


Figura 25 – A estrutura de Processos Apresentadores

A estrutura dos processos apresentadores é composta basicamente de três partes: a biblioteca de comunicação com o Gerenciador de Interface com o Usuário, o núcleo e a biblioteca de comunicação com o Gerenciador de Ativação (Figura 25).

Através da biblioteca de comunicação com o Gerenciador de Interface com o Usuário, o processo apresentador utiliza os recursos que este oferece, possibilitando a interação com o usuário e a apresentação propriamente dita de informações.

No que se refere à biblioteca de comunicação com o Gerenciador de Ativação, esta é utilizada sempre que um novo processo apresentador deve ser ativado ou quando um processo apresentador é desativado por solicitação do navegante.

Os processos apresentadores, ao serem ativados, consultam o Arquivo Hipertexto em busca das informações associadas ao nó que representam. Para isto, eles têm conhecimento da estrutura interna deste mesmo arquivo.

Para completar a descrição dos processos apresentadores, suas chamadas serão descritas a seguir.

Chamadas de Processos Apresentadores

As chamadas de processos apresentadores, que são ativados pelo Gerenciador de Ativação, têm a seguinte composição:

pa nomeDoArquivoHipertexto idNó xDaJanela yDaJanela idGA

onde:

- . **pa** é uma sequência de caracteres que representa o nome do programa que, quando ativado, se torna um processo apresentador;
- . **nomeDoArquivoHipertexto** é uma sequência de caracteres que representa o nome do arquivo hipertexto que contém as informações do nó que será apresentado;
- . **idNó** é uma sequência de caracteres que representa o identificador do nó que está associado ao processo apresentador;
- . **xDaJanela** e **yDaJanela** são dois valores inteiros que determinam a posição, no dispositivo de exibição, do canto superior esquerdo da janela associada ao processo apresentador que servirá para a apresentação propriamente dita das informações;
- . **idGA** é um valor inteiro que representa o identificador, a nível de Sistema Operacional, do Gerenciador de Ativação. O processo apresentador utiliza esta informação para estabelecer comunicação com este mesmo gerenciador através da biblioteca específica.

O Gerenciador de Interface com o Usuário

O Gerenciador de Interface com o Usuário é o processo responsável por prover recursos* que viabilizem a exibição de informações e a interação com usuários. Tanto os processos apresentadores, como o Gerenciador de Ativação solicitam tais recursos através de uma biblioteca de comunicação que é incorporada em suas estruturas.

A biblioteca de comunicação é desenvolvida a partir de uma biblioteca básica que Já se encontra disponível no ambiente de desenvolvimento do sistema (veja referência 35 na Bibliografia, página 79). Além disso, é importante lembrar que ela representa um componente importante da estrutura dos processos apresentadores e do Gerenciador de Ativação.

Os processos apresentadores utilizam os recursos oferecidos pelo Gerenciador de Interface com o Usuário, para a apresentação propriamente dita de informações do hipertexto e para interagirem com o navegador. O Gerenciador de Ativação, por sua vez, também solicita recursos para interação com navegadores.

Finalmente, vale ressaltar que a possibilidade de migração para um Gerenciador de Interface com o Usuário que ofereça recursos mais sofisticados, pertencentes a padrões modernos, deve ser sempre considerada, pois uma melhor elaboração da interface com o usuário de um sistema é uma fator determinante do sucesso operacional do mesmo.

*dispositivos lógicos de entrada e janelas para exibição de textos ou imagens são alguns exemplos desses recursos.

IV - A Fase de Implementação

IV.1 – Os Módulos de Implementação

Apartir dos requisitos funcionais estabelecidos (veja seção 1.3 – Os Requisitos Funcionais, página 37) e levando-se em consideração o modelo proposto (veja seção III.1 – O Modelo, página 44), os seguintes módulos de programação foram efetivamente implementados*¹, possibilitando o real desenvolvimento de um sistema hipertexto para navegação: GA.c, pa.c, marcador.c e comunicaMarcador.c. Estes referem-se respectivamente aos códigos fonte do Gerenciador de Ativação, do(s) processo(s) apresentador(es), do utilitário Marcador e do programa ponte (veja parágrafo 4 da seção 111.2 – A Dinâmica do Modelo, página 47) referente a este último. Além disso, foram desenvolvidas as seguintes bibliotecas de funções: liblblb_GA.a (módulo de implementação: blb_GA.c), que é incorporada na estrutura dos processos apresentadores (veja subseção Os Processos Apresentadores, página 58); libipc_srv.a (m.i.: ipc_srv.c), que é incorporada nas estruturas do Gerenciador de Ativação e do utilitário Marcador; e, por último, libipc_cli.a (m.i.: ipc_cli.c), que é incorporada na estrutura dos processos apresentadores através da biblioteca liblblb_GA.a*² e na estrutura do programa ponte comunicaMarcador.

Todas as entidades citadas no parágrafo anterior serão descritas a seguir.

. . .

Módulo de implementação: GA.c*³

Descrição: refere-se ao programa GA que está associado ao componente **Gerenciador de Ativação** do modelo proposto.

Ao ser ativado, o programa GA promove a exibição de uma janela para interação com o usuário análoga ao esquema apresentado na Figura 26. Com essa ativação, todos os Arquivos **Hipertexto** existentes no diretório corrente*⁴ se tornam disponíveis para navegação.

*1 linguagem de programação utilizada: C.

*2 funções da biblioteca liblblb_GA.a utilizam funções da biblioteca libipc_cli.a.

*3 este módulo, na implementação propriamente dita, foi dividido em dois: GA.c e GA_ui.c. Este último contém chamadas de funções e estruturas de dados necessárias para inicialização e ativação dos componentes da interface com o usuário associada ao programa GA.

*4 ou seja, no diretório onde ocorreu a ativação do programa.

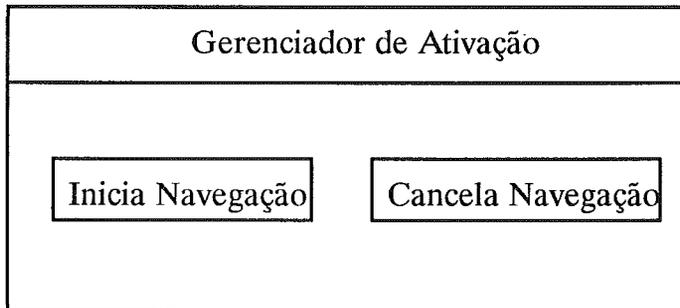


Figura 26 – Janela associada ao programa GA

Ao se pressionar o botão com rótulo "Inicia Navegação", uma nova janela surge na tela (Figura 27). Nesta, o usuário pode selecionar o **Arquivo Hipertexto** que deseja navegar. O botão com rótulo "Cancela Navegação" permite, por sua vez, que um processo de navegação seja encerrado, pois promove o cancelamento de todos os **processos apresentadores** que estejam ativos em um determinado momento.

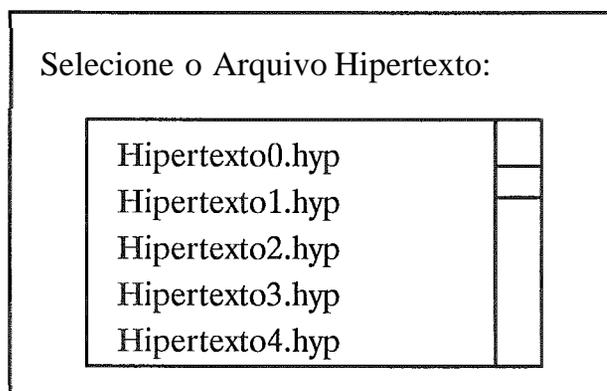


Figura 27 – Janela associada ao botão com rótulo "Inicia Navegação"

Bibliotecas utilizadas: libipc_srv.a

As funções da biblioteca **libipc_srv.a** são utilizadas no programa **GA** para o estabelecimento da comunicação entre o próprio **Gerenciador de Ativação** e os **processos apresentadores**.

. . .

Módulo de implementação: pa.c*

Descrição: refere-se ao programa pa que está associado ao componente processo apresentador do modelo proposto.

Ao ser ativado, o programa pa promove a exibição de uma janela para interação com o usuário análoga ao esquema apresentado na Figura 28. Com essa ativação, as informações associadas ao nó que o processo apresentador, em questão, representa são também apresentadas.

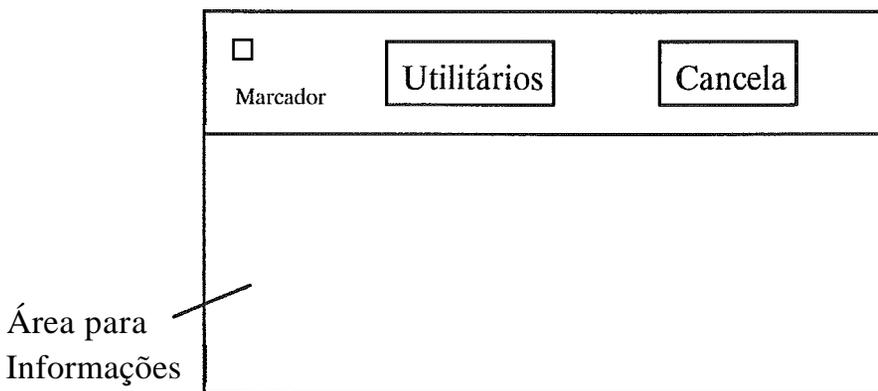


Figura 28 – Janela associada ao programa pa

Na região superior da janela para interação citada acima, existem três elementos de interface que podem ser descritos da seguinte maneira:

. o item de escolha com rótulo "Marcador"

Ao ser selecionado (Figura 29.a), provoca a execução do programa **comunicaMarcador** com a opção **-m** (veja a seguir a descrição do módulo de implementação **comunicaMarcador.c**). Ao ser desselecionado (Figura 29.b), provoca a execução do programa **comunicaMarcador** com a opção **-s** (veja a seguir a descrição do módulo de implementação **comunicaMarcador.c**).

* este módulo, na implementação propriamente dita, foi dividido em dois: pa.c e pa_ui.c. Este último contém chamadas de funções e estruturas de dados necessárias para inicialização e ativação dos componentes da interface com o usuário associada ao programa pa.



a – selecionado



b – desselecionado

Figura 29 – O ítem de escolha

. o botão com rótulo "Cancela"

Ao ser "pressionado", promove a exibição de um menu que possui duas opções (Figura 30) que podem ser selecionadas. A primeira opção do menu ("Cancela esta Janela de Apresentação") implica no cancelamento do **processo apresentador** em questão. A segunda opção ("Cancela até a Última Apresentação Marcada") implica na execução do programa **comunicaMarcador** com a opção **-r** (veja a seguir a descrição do módulo de implementação **comunicaMarcador.c**).

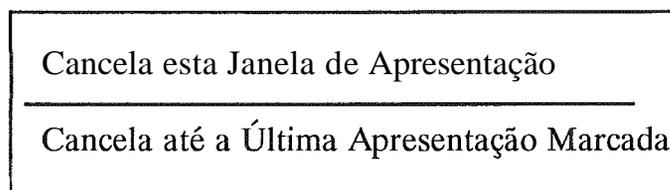


Figura 30 – Menu associado ao botão com rótulo "Cancela"

. o botão com rótulo "Utilitários"

Ao ser "pressionado", promove a exibição de uma nova janela na tela (Figura 31) que contém um lista de opções (rótulos) que, ao serem selecionadas, implicam na execução de uma chamada de programa correspondente. Tanto os rótulos exibidos na janela apresentada, como as chamadas de programas associadas, são extraídas, pelo(s) processo(s) apresentador(es), de um arquivo que, para isto, possui o seguinte formato:

nChamadas

```
{
  rótulo
  chamadaDePrograma
}nChamadas
```

onde:

- . nChamadas é um número inteiro que representa o número de chamadas de programa que são descritas no arquivo;
- . rótulo é uma sequência de caracteres que representa um rótulo que será exibido na lista de opções descrita acima;
- . chamadaDePrograma é uma sequência de caracteres que representa uma chamada de programa que será executada caso o rótulo associado seja selecionado.

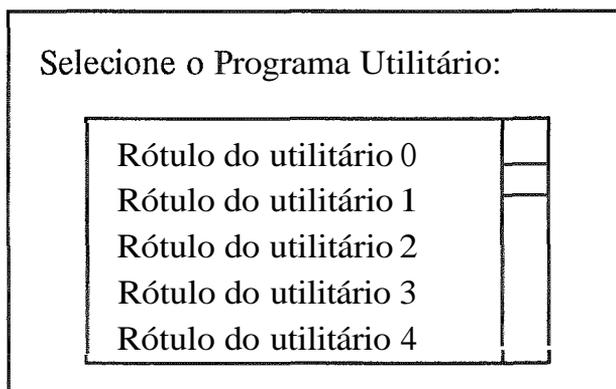


Figura 31 – Janela associada ao botão com rótulo "Utilitários"

Convenciona-se chamar este arquivo de ”**utilitários.utl**”. O mesmo deve estar presente no diretório no qual os processos apresentadores de uma navegação sejam ativados.

Bibliotecas utilizadas: libblb_GA.a

As funções da biblioteca **libblb_GA.a** são utilizadas pelo programa **pa** para o estabelecimento da comunicação entre o(s) próprio(s) **processo(s) apresentador(es)** e o **Gerenciador de Ativação** (veja subseção Os Processos Apresentadores, página 58).

. . .

Módulo de implementação: marcador.c

Descrição: refere-se ao programa **marcador** que está associado ao componente **utilitário Marcador** do modelo proposto.

Bibliotecas utilizadas: libipc_srv.a

As funções da biblioteca **libipc__srv.a** são utilizadas pelo programa **marcador** para o estabelecimento da comunicação entre o próprio **utilitário Marcador** e o **programa ponte comunicaMarcador** (veja a seguir).

. . .

Módulo de implementação: comunicaMarcador.c*

Descrição: refere-se ao programa **comunicaMarcador** que está associado ao componente **programa ponte do utilitário Marcador** do modelo proposto.

O programa **comunicaMarcador**, ao ser ativado, deve receber um dos seguintes parâmetros pela linha de comando:

parâmetro	função
-a	informar ao utilitário Marcador que um processo apresentador foi ativado.
-d	informar ao utilitário Marcador que um processo apresentador foi desativado.

* este módulo, na implementação propriamente dita, foi dividido em dois: **comunicaMarcador.c** e **blb_Marcador.c**. Este último utiliza funções da biblioteca **libipc_cli.a**.

- m informar ao utilitário Marcador que um processo apresentador foi "marcado".
- s informar ao utilitário Marcador que um processo apresentador foi "desmarcado".
- r solicitar ao utilitário Marcador o cancelamento de todos os processos apresentadores que foram ativados após aquele que foi, por último, "marcado".

Bibliotecas utilizadas: blbipc_cli.a

As funções da biblioteca **libipc_cli.a** são utilizadas pelo programa **comunicaMarcador** para o estabelecimento da comunicação entre o próprio programa **ponte comunicaMarcador** e o utilitário **Marcador**.

. . . .

Módulo de implementação: blb_GA.c

Descrição: contém funções que são utilizadas nos módulos associados aos **processos apresentadores**. Estas funções são utilizadas para o estabelecimento da comunicação entre o **Gerenciador de Ativação** e os próprios **processos apresentadores** (veja subsecção Os Processos Apresentadores, página 58). Refere-se à biblioteca **libblb_GA.a**.

Bibliotecas utilizadas: libipc_cli.a

As funções da biblioteca **libipc_cli.a** são utilizadas pelas funções descritas no módulo de implementação **blb_GA.c**. É deste modo que é estabelecida efetivamente a comunicação entre os **processos apresentadores** e o **Gerenciador de Ativação**.

. . . .

Módulo de implementação: ipc_srv.c*

Descrição: contém funções que são utilizadas nos módulos de implementação que estão associados a programas que, quando ativados, cumprem o papel de **processos servidores**. Refere-se à biblioteca **libipc_srv.a**.

Bibliotecas utilizadas: ---

* **ipc_srv** vem de *interprocess communication* e servidor.

. . .

Módulo de implementação: ipc_cli.c*

Descrição: contém funções que são utilizadas nos módulos de implementação que estão associados a programas que, quando ativados, cumprem o papel de **processos** clientes. Refere-se à biblioteca **libipc_cli.a**.

Bibliotecas utilizadas: ---

. . .

Informações Complementares

Vale acrescentar que para a efetiva utilização do recurso **Marcador**, especificado neste projeto, o programa pa, ao ser executado, deve encontrar obrigatoriamente no arquivo de inicialização associado (veja seção I11.2 – A Dinâmica do Modelo, página 47) uma chamada do **programa ponte comunicaMarcador** com a opção **-a** (veja a descrição do módulo de implementação **comunicaMarcador.c**, página 66). De forma complementar, o programa pa, ao ser cancelado, deve encontrar no arquivo de **finalização** associado (veja seção I11.2 – A Dinâmica do Modelo, página 47) uma chamada do **programa ponte comunicaMarcador** com a opção **-d** (veja a descrição do módulo de implementação **comunicaMarcador.c**, página 66).

. . .

IV.2 – Considerações sobre a Incorporação de Novos Utilitários

Para exemplificar a utilização do mecanismo de extensibilidade proposto no projeto deste sistema (veja parágrafo 4 da seção I11.2 – A Dinâmica do Modelo, página 47), basta considerar as etapas que devem ser cumpridas na incorporação do utilitário Histórico, associado ao recurso do processo de navegação de mesmo nome (veja seção 1.2 – Recursos do Processo de Navegação, página 32):

* ipc_cli vem de *interprocess communication e cliente*.

a. o desenvolvimento dos módulos de implementação associados ao próprio utilitário Histórico e ao seu respectivo programa ponte. Este último deve possibilitar, através de opções de execução, o informe das seguintes situações ao utilitário Histórico: a ativação e o cancelamento de processo(s) apresentador(es). Além disso, ele deve possibilitar também a solicitação do recurso Histórico em si (por exemplo, a apresentação de uma janela que exiba uma "lista cronológica" de processos apresentadores ativos);

b. a incorporação nos arquivos de inicialização e de **finalização** utilizados pelos processos apresentadores das chamadas do programa ponte com as opções respectivas de informe de ativação e informe de cancelamento;

c. incorporação no arquivo "utilitários.utl" (veja explicação sobre "o botão com rótulo "Utilitários" ", página 65) da chamada do programa ponte com a opção de solicitação do recurso Histórico.

IV.3 - O Recurso Comentário

A idéia de se discutir, em seção exclusiva, o recurso Comentário provém da importância que o mesmo exerce no cumprimento do objetivo primordial do projeto deste sistema hipertexto: a integração de um grupo de pesquisa. Deste modo, será proposto, a seguir, um mecanismo que viabilize a realização de comentários públicos, ou seja, de comentários que poderão ser compartilhados por todos os usuários.

. . .

Cada hipertexto estará associado a um arquivo que armazenará referências sobre comentários realizados por usuários. Este arquivo deverá se encontrar no mesmo diretório do arquivo hipertexto e terá seu mesmo nome, porém com terminação diferente: ".RCMP" (referências a comentários pessoais). Ele deverá ter também acesso irrestrito para escrita e leitura; todos poderão acessar e acrescentar informações.

Dentro de um arquivo de referências a comentários estarão relacionados os nomes dos usuários que fizeram comentários sobre o hipertexto correspondente, juntamente com os nomes dos arquivos que armazenam os comentários em si. Estes Últimos, com terminação ".CMT" (comentário), estarão localizados em diretórios dos próprios usuário (Figura 32).

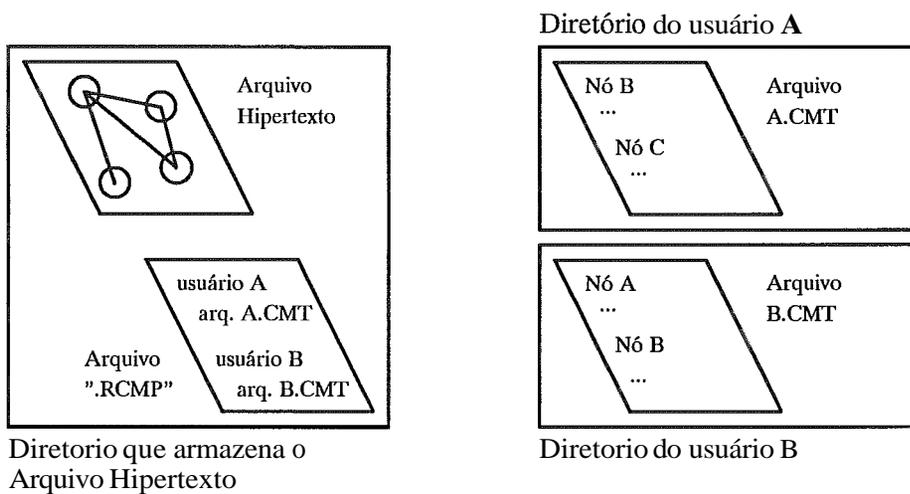


Figura 32 – Os Arquivos do recurso Comentário

No arquivo ".CMT", estará discriminado, para cada nó do hipertexto, o comentário feito pelo usuário. Este poderá ser analisado quando, em um processo de navegação, o nó em questão estiver sendo apresentado a qualquer usuário.

Todo o mecanismo acima descrito se tornará viável por intermédio do desenvolvimento de um programa utilitário que, durante o processo de navegação, se encarregará de registrar e apresentar comentários quando os usuários assim desejarem.

IV.4 – Características do Ambiente de Implementação

- . equipamento disponível: estações gráficas Sun;
- . Sistema Operacional do ambiente: SunOs (UNIX/BSD), versão 4.1;
- . bibliotecas de comunicação e núcleo de programas foram desenvolvidos na linguagem de programação C. Nas bibliotecas, foram utilizadas chamadas ao sistema (*system calls*) que operacionalizam mecanismos de IPC (*Interprocess Communication*) do Sistema Operacional;
- . Gerenciador de Interface com o Usuário: OpenWindows, versão 2.0.

IV.5 – Formatos Padrão do Sistema

No sistema hipertexto implementado, três tipos de informação podem ser apresentadas: textos, imagens e programas. Isto pressupõe a existência de formatos padrão de armazenamento que, neste caso, são os seguintes:

- . informações textuais são diretamente reconhecidas pelo sistema se estiverem armazenadas em arquivos de caracteres ASCII;
- . imagens são diretamente reconhecidas pelo sistema se estiverem armazenadas em arquivos do tipo Sun *rasterfile* (veja referência 37 na Bibliografia, página 79);
- . programas devem estar em código executável, compatível com o Sistema Operacional SunOs, versão 4.1.

Por último, vale lembrar que informações armazenadas em outros formatos podem ser apresentadas através da utilização de filtros.

Conclusões

A Implementação

Este trabalho resultou na implementação de um sistema hipertexto de consulta; todos os programas desenvolvidos compreendem um total de mais de 10.000 (dez mil) linhas de programação na linguagem C. Foram gastos 6 (seis) meses na Fase de Implementação que foi realizada pelo próprio autor.

A maior dificuldade encontrada durante a Implementação foi, certamente, a depuração dos programas que estavam sendo desenvolvidos. Esta é fruto da decisão de se utilizar, como fundamento básico no projeto de desenvolvimento do sistema, o conceito de processo do Sistema Operacional (veja seção III.1 – O Modelo, página 41). Por exemplo, nos casos em que o sistema reagia de maneira não prevista, era necessária a checagem de diversos módulos de implementação (dos diversos processos) com o objetivo de se descobrir um eventual engano cometido.

Com exceção dos recursos Histórico, Comentário, Impressão, Mapas, Dicas e Sugestões todos os demais requisitos funcionais do sistema (veja seção I.3 – Os Requisitos Funcionais, página 37) foram atendidos. Foi desenvolvido também um programa utilitário (Hyperedit (veja Apêndice A, página 80)) que pode ser considerado com um protótipo inicial de um sistema hipertexto de autoria que convenientemente deve ser implementado.

O sistema hipertexto de navegação desenvolvido foi chamado de Sistema Apoio (veja Apêndice B, página 87).

O Modelo Utilizado

Neste trabalho, o elemento de maior destaque é o modelo de implementação proposto (veja seção III.1 – O Modelo, página 44). Com ele, a incorporação dos recursos do processo de navegação que, mesmo especificados, não foram implementados se dá, por exemplo, com o desenvolvimento de programas específicos, sem que seja necessário qualquer alteração nos códigos já implementados. Isto demonstra a flexibilidade que se tem, com o uso do modelo proposto, em relação à extensibilidade funcional do sistema.

A possível escassez de recursos computacionais* que poderia advir com o emprego do modelo proposto não chegou a ser considerada. Primeiro, porque ela não ocorreu. Além disso, vale lembrar que, a cada dia, novos e mais sofisticados equipamentos vem sendo desenvolvidos e, portanto, utilizados em ambientes computacionais.

* problemas como falta de memória, sobrecarga de processamento etc.

Como diretriz básica de possíveis trabalhos que possam ser efetuados como prosseguimento a este, cabe a seguinte pergunta: Que outros recursos de um sistema hipertexto podem (ou não) ser incorporados ao sistema desenvolvido com o mecanismo de extensibilidade projetado?

A especificação do modelo proposto e sua efetiva implementação foram a grande motivação durante o desenvolvimento deste trabalho.

Considerações Finais

Neste trabalho, foi apresentado o projeto de desenvolvimento de um sistema hipertexto para estações gráficas em ambiente UNIX. Este sistema será utilizado inicialmente para a integração dos documentos acadêmicos e administrativos que são produzidos pelo grupo de interesse em Computação Gráfica da COPPE, ou seja, ele tem o objetivo de apoiar o trabalho de pesquisa desse mesmo grupo.

Uma das preocupações constantes durante o projeto foi a de explorar, de modo geral, os recursos do ambiente de operação do sistema. O resultado disto foi a natural convergência, no projeto de implementação, para uma arquitetura do tipo cliente-servidor, que vem se tornando cada vez mais difundida em ambientes UNIX.

Em relação ao emprego dos conceitos de hipertexto como fundamento básico no projeto, espera-se que esta premissa seja decisiva ao se pretender desenvolver um sistema, de fácil uso, para integrar documentos diversos. Além disso, o sistema desenvolvido pode ser considerado como plataforma inicial para estudos futuros na área de hipertexto.

Bibliografia

- [1] SHNEIDERMAN, B. e KEARSLEY, G.
Hypertext Hands-On
Addison-Wesley Publishing Company, 1989.
- [2] CONKLYN, J.
Hypertext: An Introduction and Survey
IEEE Computer (20,9), setembro de 1989, pgs. 17 - 41.
- [3] YANKELOVICH, N., HAAN, B. J. e DRUCKER, S. M.
Intermedia: The Concept and the Construction of a Seamless Information Environment
IEEE Computer (21,1), janeiro de 1988, pgs. 81 - 96.
- [4] WILLIAMS, G.
Hypercard
Byte, dezembro de 1987, pgs. 109 - 117.
- [5] BUSH, V.
As We May Think
Atlantic Monthly (176,1), julho de 1945, pgs. 101 - 108.
- [6] ENGELBART, D.
A Conceptual Framework for the Augmentation of Man's Intellect
Vistas in Information Handling (1), 1963.
Sparts Books, Londres.
- [7] MEYROWITZ, N.
Intermedia: The Architecture and Construction of an Object-Oriented
Hypermedia System and Applications Framework
Proc. OOPSLA' 86 Conf., pgs. 186 - 201.
- [8] HALASZ, F.
Reflections on Notecards: Seven Issues for the Next Generation of
Hypermedia Systems
Communications of the ACM (31,7), julho de 1988, pgs. 836 - 852.

- [9] AKSCYN, R. M., MCCRACKEN, R. L. e YONDER E. A.
KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations
Communications of the ACM (31,7), julho de 1988, pgs. 820 – 835.
- [10] ROBERTSON, C. K., MCCRACKEN, D. e NEWELL, A.
The ZOG Approach to Man–Machine Communication
International Journal of Man–Machines Studies (14), 1981, pgs. 461 – 488.
- [11] SHNEIDERMAN, B.
User Interface Design for the Hyperties Eletronic Encyclopedia
Proc. ACM Hypertext' 87 Conf., pgs. 189 – 194.
- [12] DELISLE, N. e SCHWARTZ, M.
Neptune: A Hypertext System for CAD Applications
Proc. ACM SIGMOD' 86, pgs. 132 – 142.
- [13] APPLE COMPUTER
Hypercard Stack Design Guidelines
Addison–Wesley Publishing Company, 1989.
- [14] BROWN, P. J.
Turning Ideas into Products: The Guide System
Proc. ACM Hypertext' 87 Conf., pgs. 33 – 40.
- [15] RAYMOND, D. R. e TOMPA, F. W.
Hypertext and the New Oxford English Dictionary
Communications of the ACM (31,7), julho de 1988, pgs. 871 – 879.
- [16] FEINER, S., NAGY, S. e VAN DAM, A.
An Experimental System for Creating and Presenting Interactive Graphical Documents
ACM Trans. Graphics (1,1), janeiro de 1982, pgs. 59 – 77.

- [17] BIGELOW, J. e RILEY, V.
Manipulating Source Code in Dynamic Design
Proc. ACM Hypertext' 87 Conf., pgs. 397 – 408.
- [18] BIGELOW, J.
Hypertext and CASE
IEEE Software (5,2), março de 1988, pgs. 23 – 27.
- [19] CONKLIN, J. e BEGEMAN, M. L.
gIBIS: A Hypertext Tool for Exploratory Policy Discussion
ACM Trans. Office Information Systems (6,4), outubro de 1988,
pgs. 303 – 331.
- [20] KRENDL, K. A. e LIEBERMAN, D. A.
Computer and Learning: A Review of Recent Research
Journal of Educational Computing Research (4,4), 1988, pgs. 367 – 389.
- [21] KOZMA, R. B.
The Implications of Cognitive Psychology for Computer-Based Learning
Tools
Educational Technology, novembro de 1987, pgs. 20 – 25.
- [22] CAMPBELL, B. e GOODMAN, J. M.
HAM: A General Purpose Hypertext Abstract Machine
Communications of the ACM (31,7), julho de 1988, pgs. 856 – 861.
- [23] AKSCYN, R. e outros.
Interchanging Hypertexts
Proc. ACM Hypertext' 89 Conf., pgs. 379–381.
- [24] NIELSEN, J.
The Art of Navigating Through Hypertext
Communications of the ACM (33,3), março de 1990, pgs. 297 – 310.
- [25] Grupo de Informática Aplicada ao Ensino (GINAPE) / NCE-UFRJ
Notas pessoais, 1991.

- [26] NIELSEN, J.
Hypertext and Hypermedia
Academic Press Inc., 1990.
- [27] BERNSTEIN, M.
The Bookmark and the Compass: Orientation Tools for Hypertext Users
ACM SIGOIS Bulletin (9,4), outubro de 1988, pgs. 34 – 45.
- [28] PRESSMAN, R. S.
Software Engineering – A Practitioner's Approach
McGraw–Hill Book Company, 1987.
- [29] BOOCH, G.
Object Oriented Design with Applications
The Benjamin/Cummings Publishing Company, 1991.
- [30] STEVENS, W. R.
UNIX Networking Programming
Prentice Hall Software Series, 1990.
- [31] HAVILAND, K., e SALAMA, B.
UNIX System Programming
Addison–Wesley Publishing Company, 1987.
- [32] SHNEIDERMAN, B.
Designing the User Interface: Strategies for Effective Human–Computer
Interaction
Addison–Wesley Publishing Company, 1987.
- [33] SUN MICROSYSTEMS
SunOs Reference Manual
1987
- [34] SUN MICROSYSTEMS
OpenWindows User's Guide (Versão 2.0)
1990

- [35] HELLER, R.
XView Programming Manual
O'Reilly & Associates, 1990.
- [36] SUN MICROSYSTEMS
OpenWindows Developer's Guide (Versão 1.1)
1990
- [37] SUN MICROSYSTEMS
Pixrect Reference Manual (Versão 4.1)
1990
- [38] SUN MICROSYSTEMS
OPEN LOOK Graphical User Interface Application Style Guidelines
1990

Apêndice A: Hyperedit - Tutorial

Este tutorial fornece as informações necessárias para a utilização do programa utilitário **Hyperedit** que pertence ao **Sistema Apoio**. Como requisito básico para essa utilização, considera-se o conhecimento de operação do utilitário **Guide***1*2, integrante do **Sistema OpenWindows***1. As seguintes seções compõe este tutorial:

1. Criando um Hipertexto
2. O Funcionamento do Teclado
3. Outros Documentos

. . .

1 - Criando um Hipertexto

Neste seção, será criado um hipertexto composto de dois nós e que possui a estrutura vista na Figura 1:

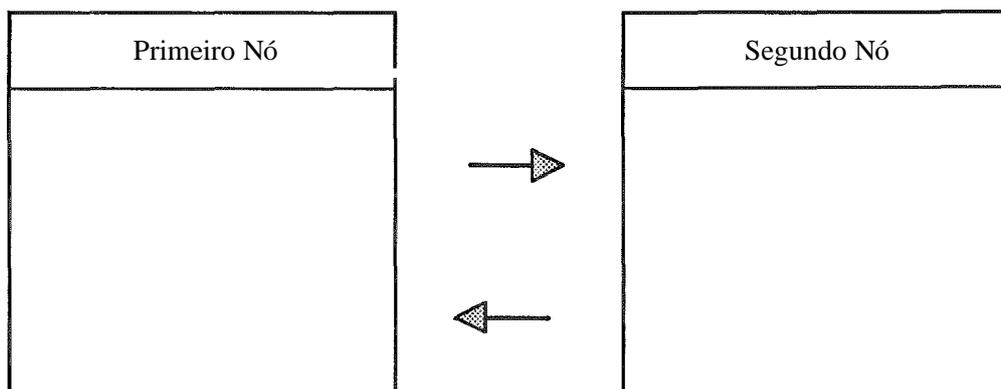


Figura 1

*1 **Guide** e **OpenWindows** são marcas registradas da Sun Microsystems.

*2 para maiores informações sobre o utilitário **Guide**, veja seção 3, referência a.

1.1 – Criando os Arquivos de *Layout*

a. utilize o utilitário Guide para criar um arquivo de especificação de interface (**teste0.G**) que deve conter: a descrição de uma *base window* (500,300) e a descrição de uma *text window* ((0,0);(500,300))*.

b. utilize o utilitário Guide para criar um arquivo de especificação de interface (**teste1.G**) que deve conter: a descrição de uma *base window* (500,300), a descrição de uma *text window* ((0,0);(500,150)), a descrição de uma *canvas window* ((0,150);(250,150)) e a descrição de uma *term window* (250,150);(250,150)).

c. execute as seguintes linhas de comando para a criação dos arquivos **de layout teste0.lyt** e **teste1.lyt**:

```
geraArquivoDeLayout testeO.G teste0.lyt
```

```
geraArquivoDeLayout teste1.G teste1.lyt
```

1.2 – Editando o Hipertexto teste.hyp

1.2.1 – Ativando o Utilitário Hyperedit

Execute a seguinte linha de comando para editar o hipertexto **teste.hyp**:

```
hyperedit teste.hyp
```

Este procedimento resultará na abertura de uma janela de controle principal análoga ao esquema apresentado na Figura 2.

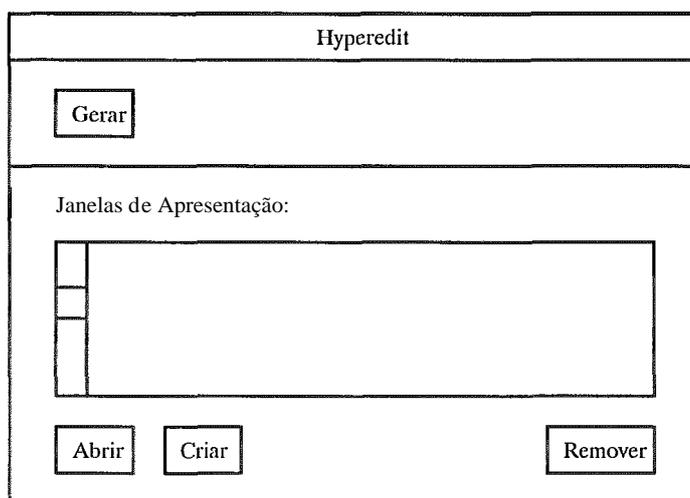


Figura 2 – A janela de controle principal

* ((origem);(dimensões)).

1.2.2 – Criando os Nós do Hipertexto

a. "Pressione" o botão da janela de controle principal com rótulo "**Criar**". Digite "teste0.lyt" no campo com rótulo "**Arquivo com Layout:**" da nova janela que será apresentada. "Pressione" o botão com rótulo "**Criar**" desta última janela.

O primeiro nó (**J0**) está criado e sua respectiva **Janela de Apresentação** será exibida (Figura 3).

b. "Pressione" o botão da janela de controle principal com rótulo "**Criar**". "Pressione" o botão com rótulo "**Não Confirma**" da janela que será exibida com a pergunta: "**Confirma o layout do arquivo "teste0.lyt" para a nova Janela?**". Digite "teste1.lyt" no campo com rótulo "**Arquivo com Layout:**" da nova janela que será apresentada. "Pressione" o botão com rótulo "**Criar**" desta última janela.

O segundo nó (**J1**) está criado e sua respectiva **Janela de Apresentação** será exibida (Figura 3).

1.2.3 – Rotulando os Nós do Wipertexto

a. "Pressione" o botão com rótulo "**Propriedades**" da **Janela de Apresentação** associada ao primeiro nó (**J0**) criado na etapa anterior. Digite "**Primeiro Nó**" no campo com rótulo "**Rótulo:**" da janela que será apresentada. "Pressione" o botão com rótulo "**Atualizar Props.**" deste última janela.

b. "Pressione" o botão com rótulo "**Propriedades**" da **Janela de Apresentação** associada ao segundo nó (**J1**) criado na etapa anterior. Digite "**Segundo Nó**" no campo com rótulo "**Rótulo:**" da janela que será apresentada. "Pressione" o botão com rótulo "**Atualizar Props.**" deste última janela.

1.2.4 – Inserindo as Informações nos Nós

a. Na **Janela de Apresentação** associada ao nó **J0** efetue a seguinte operação:
 – Digite " < **Vá para o Nó J1** > **J1** > " na área para apresentação de informações (Figura 3). Os três últimos caracteres da sequência a ser digitada ("J1 > ") não serão exibidos quando o hipertexto que está sendo editado for navegado. Eles são utilizados, porém, para indicar que o texto " < **Vá para o Nó J1** > " está associado ao nó **J1**.

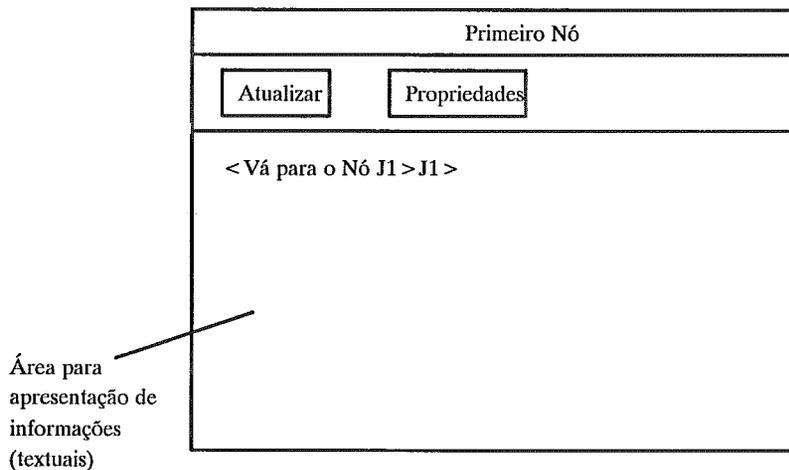


Figura 3

b. Na Janela de Apresentação associada ao nó **J1**, efetue a seguinte operação:

- Digite "< Vá para o Nó **J0** > **J0** >" na área para apresentação de informações textuais (Figura 4);

- Digite, no campo com rótulo "Arquivo:" da área para apresentação de imagens (rótulo "Área para Imagem:"), o nome de um arquivo **Sun rasterfile***¹ (Figura 4);

- Digite, no campo com rótulo "Chamada:" da área para execução de programas (rótulo "Área para Programa:"), a sequência "ls" (Figura 4). Esta Última refere-se ao comando ls do Sistema Operacional UNIX*².

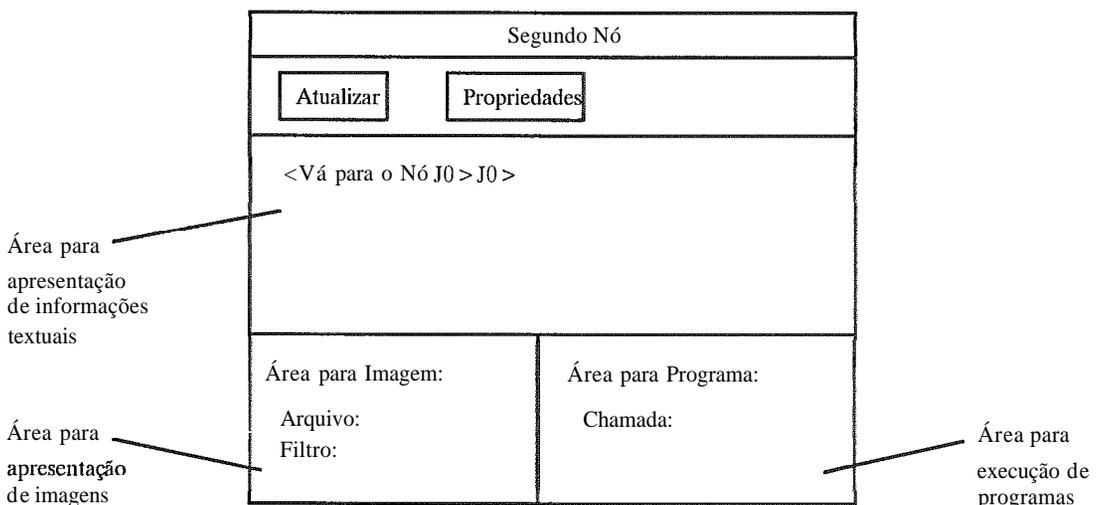


Figura 4

*1 para maiores informações a respeito de arquivos Sun *rasterfile* veja seção 3, referência b.

*2 para maiores informações a respeito de comandos do Sistema Operacional UNIX veja seção 3, referência c.

1.2.5 – Gerando o Hipertexto teste.hyp

a. "Pressione" o botão com rótulo "Atualizar" da Janela **de Apresentação** associada ao nó **J0**. "Pressione" o botão com rótulo "Atualizar" da janela que será exibida com a pergunta: "Deseja realmente Atualizar?".

Efetue as mesmas operações descritas anteriormente para o nó **J1**.

b. "Pressione" o botão com rótulo "Gerar" da janela de controle principal. "Pressione" o botão com rótulo "Continua" da janela que será exibida com a exclamação: **"ATENÇÃO! Para uma correta Geração, todas as Janelas devem estar Atualizadas!"**.

1.2.6 – Encerrando a Edição

No menu associado à janela de controle principal, selecione a opção **"Quit"**. "Pressione" o botão com rótulo "Terminar" da janela que será exibida com a seguinte pergunta: "Deseja realmente Terminar?".

. . .

O hipertexto **teste.hyp** já pode ser utilizado para navegação.

. . .

Na janela de controle associada ao utilitário Hyperedit, haviam dois botões que não foram utilizados na edição do hipertexto **teste.hyp**: o botão com rótulo **"Abrir"** e o botão com rótulo "Remover".

O botão com rótulo "Abrir" é utilizado para a abertura de uma **Janela de Apresentação** que tenha sido previamente selecionada da lista existente na própria janela de controle.

O botão com rótulo "Remover" é utilizado para a remoção de uma **Janela de Apresentação** que tenha sido previamente selecionada da lista existente na própria janela de controle. Neste caso, todas as informações vinculadas ao nó associado a uma Janela removida são perdidas (!).

2 – O Funcionamento do Teclado

Os Caracteres Delimitadores de Chaves de Navegação

Os caracteres delimitadores de **chaves de** navegação são obtidos pressionando-se as teclas referentes aos caracteres ”<” e ”>”.

Caracteres Acentuados

A obtenção de caracteres acentuados é realizada de modo análogo ao em máquinas de escrever convencionais. Por exemplo, para se obter o carácter ”á”, deve se pressionar inicialmente a tecla referente ao carácter ”’” e, em seguida, àquela referente ao carácter ”a”.

O Caractere ”Ç”

A obtenção do carácter ”ç” é realizada de modo análogo ao em máquinas de escrever convencionais, ou seja, através da tecla localizada à direita da tecla referente ao carácter ”l”.

A Tecla ESC

Sempre que for necessária a obtenção do carácter originariamente associado a uma tecla, utilize a tecla ESC, pressionando-a anteriormente. Por exemplo, para se obter o carácter ”<”, pressione a tecla ESC e, em seguida, a tecla referente ao próprio carácter ”<”. Vale lembrar que a tecla associada ao carácter ”<” havia sido utilizada para outros fins .

3 – Outros Documentos

- a. OpenWindows Developer's Guide (Versão 1.1)
Sun Microsystems, 1990.
- b. Pixrect Reference Manual (Versão 4.1)
Sun Microsystems, 1990.
- c. SunOs Reference Manual
Sun Microsystems, 1987.

Apêndice B: Sistema Apoio (versão 1.0)

Guia de Instalação

Este guia fornece as informações necessárias para a instalação do Sistema Apoio. Para isto, ele é dividido nas seguintes seções:

1. Requisitos para instalação
2. Procedimento de instalação
3. Preparando o ambiente de ativação

. . .

1 – Requisitos para instalação

- a. o sistema operacional do ambiente deve ser o SunOS* versão 4.0 ou compatível;
- b. o software OpenWindows* versão 2.0 ou compatível deve ter sido previamente instalado;
- c. devem existir, pelo menos, 1.500 kbytes de espaço em disco disponíveis.

2 – Procedimento de instalação

Os arquivos fonte do Sistema Apoio se encontram originariamente armazenados em um único disquete (3 1/2 ”). Deste modo, as seguintes etapas devem ser cumpridas para a sua instalação:

- a. após introdução do disquete na unidade de leitura correspondente, dirija-se ao diretório sobre o qual o diretório referente ao Sistema Apoio será criado;
- b. extraia os arquivos existentes no disquete através da execução da seguinte linha de comando:

```
tar xvf arquivoReferenteAUnidadeDeLeitura
```

* SunOs e OpenWindows são marcas registradas da Sun Microsystems.

3 – Preparando o ambiente de ativação

Para a efetiva utilização do sistema, as seguintes etapas devem ser cumpridas:

a. completar os arquivos *.login* dos futuros usuários do sistema com os seguintes ítems:

```
setenv APOIOHOME pathnameDoDiretorioDoSistemaApoio
set path = ($path $APOIOHOME/bin)
```

b.1 criar no diretório `OPENWINHOME/lib/fonts` o seguinte arquivo link*:
`hyperfont.fb -> $APOIOHOME/lib/fonts/hyperfont.fb`.

b.2 executar, no diretório `OPENWINHOME/lib/fonts`, o comando *bldfamily*.
 Executar a seguinte linha de comando: *xset fp rehash*.

c. dirija-se ao diretório referente ao Sistema Apoio e execute o "comando" *makesis* para a criação dos executáveis do sistema.

. . .

O Sistema Apoio já pode ser utilizado.

* a variável do environment APOIOHOME já deve ter sido, portanto, criada (vide ítem a.).