

Hiper Autor: Um Método para a Especificação de Aplicações em Hipermissão

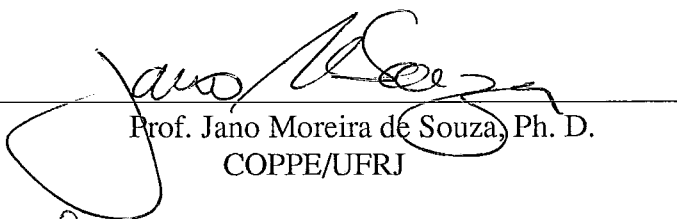
Karin Koogan Breitman

Tese submetida ao Corpo Docente da Coordenação dos Programas de Pós-Graduação em 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.

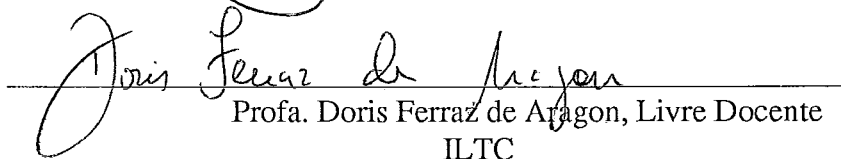
Aprovada por:



Profa. Ana Regina Cavalcanti da Rocha, D. Sc.
COPPE/UFRJ
(Presidente)



Prof. Jano Moreira de Souza, Ph. D.
COPPE/UFRJ



Profa. Doris Ferraz de Aragon, Livre Docente
ILTC



Profa. Neide dos Santos, Mestre em Educação
COPPE/UFRJ

Rio de Janeiro - RJ - Brasil
Setembro de 1993

BREITMAN, Karin Koogan

HIPER AUTOR: Um método para a especificação de sistemas de Hipermissão

[Rio de Janeiro] 1993

xii, 134 pg., 29,7 cm. (COPPE/UFRJ, M.Sc. Engenharia de Sistemas e Computação, 1993)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Ambientes de desenvolvimento de software

I. COPPE/UFRJ II. Título (série)

Esta tese é dedicada a:

Angela Maria Bouth

Agradecimentos:

A amiga e orientadora, Ana Regina, pela dedicação, carinho, "força" e pelas broncas também.

À querida Neide, que não só ajudou na orientação, mas acreditou na idéia desde o começo.

Ao André, pelo TLB e pela força que deve ter feito para deixar o computador só para mim.

Aos meus irmãos, Evinha, Edu e Marcelo pela força (bruta às vezes).

À minha família, pelo carinho e apoio e preocupação.

À Aida, por ter segurado a(s) barra(s).

A todos os Barbosa, (de Oliveira ou nem tanto) pela ajuda incondicional ao longo do trabalho.

À Lygia, pelo carinho, apoio moral e sábios conselhos.

À Gilda, pela atenção e incansáveis ouvidos.

À Fernanda, pelo companheirismo e atenção.

À Claudinha e Ana Paula, pela paciência infinita.

Ao prof. Daniel Schwabe, pela ajuda na pesquisa bibliográfica.

Ao John, pelo apoio moral e cafezinho (sss).

Ao Xexéo, pelos livros e reflexões .

Ao Jano, pelas idéias e o chá que acompanhava.

À Vera, pela gentileza.

À CAPES, pelo apoio financeiro.

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

**HIPER AUTOR: Um método para a especificação de
aplicações do tipo Hipermídia**

Karin Koogan Breitman

Setembro de 1993

Orientadora: Ana Regina Cavalcanti da Rocha, D. Sc.

Programa: Engenharia de Sistemas e Computação

A disseminação dos sistemas do tipo hipertexto/hipermídia tem trazido à tona as dificuldades relacionadas à autoria deste tipo de sistema. Para minimizar esta problemática, propomos o Hiper Autor, um método para a especificação e projeto de aplicações hipermídia que tem como objetivo principal apoiar uma equipe composta por profissionais de diversas áreas neste processo. Para tal, o Hiper Autor se utiliza de técnicas de elicitação de requisitos e modelagem, de maneira a fornecer um conjunto de diretivas, organizando o pensamento e o trabalho da equipe ao longo do processo de desenvolvimento do software.

O método consiste na modelagem da base de dados da aplicação, através de uma linguagem gráfico-linguística, representada por um conjunto de formulários. Esta representação, além de gerar documentação do projeto, e, portanto, possibilitar manutenções futuras, permite que a qualidade da aplicação possa ser avaliada em vários estágios de seu desenvolvimento.

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

**HIPER AUTOR: A method for specification and
project of hypermedia applications**

Karin Koogan Breitman

September, 1993

Thesis Supervisor: Ana Regina Cavalcanti da Rocha, D. Sc.

Department: Systems Engineering and Computer Science

The widespread use of hypertext and hypermedia systems has brought up the difficulties related to their authoring. To minimize this problem we propose Hiper Autor, a method for the specification and project of hypermedia applications which has, as its main objective, to support a team made up of professionals from the many fields encompassed by the project. For this Hiper Autor uses techniques for the elicitation of the system's requirements so as to supply a set of directives which organizes the thinking process and actual work of the team during the software's development.

The method consists in modeling the application's database through a graphical-linguistical language represented by a set of forms. This representation allows not only the documentation of the project and its future maintenance, but also its evaluation at several phases of its development.

Sumário

CAPÍTULO 1:

Introdução..... 1

1.1 Justificativas 1

1.2 Objetivos da Tese 4

1.3 Organização da Tese 5

CAPÍTULO 2:

Hipertexto/Hipermídia 7

2.1 Introdução 7

2.2 Definição 7

2.3 Conceitos Básicos de Hipertexto 9

2.4 Autoria de sistemas do tipo hipertexto/hipermídia 12

2.4.1 A metáfora de Livro Eletrônico 15

2.4.2 A Metáfora da Excursão 16

2.4.3 Exemplo de uma aplicação construída a partir da metáfora de livro eletrônico: O Hipertexto Hands On! 17

2.4.4 Exemplo de um aplicativo construído a partir da metáfora de excursão: Guia das Oportunidades da Arqueologia Voluntária..... 19

2.5 Arquitetura Básica dos sistemas de Hipertexto/ /Hiperídia 22

2.6 Conclusões 26

CAPÍTULO 3:

O desenvolvimento de aplicações hiperídia .. 28

3.1 Introdução 28

3.2 Métodos de autoria para sistemas hiperídia 31

3.2.1 Métodos de autoria dependentes de ferramentas 32

3.2.1.1 g-IBIS [Conk 87]	32
3.2.1.2 KMS [Aksc 88].....	34
3.2.1.3 HELENA [Petr 93].....	35
3.2.2 Métodos de autoria independentes de ferramentas	37
3.2.2.1 As diretrizes propostas por Ernest Perez	37
3.2.2.2 O método ThyDoc [Sobi 91].....	39
3.2.2.3 O método HDM [Garz 90], [Garz 91].....	42
3.3 Avaliação da Qualidade	46
3.3.1 Avaliação da qualidade de ferramentas para implementação de sistemas de hipertexto	46
3.3.2 Avaliação da Qualidade de aplicações do tipo Hipertexto/Hípermedia	49
3.4 Conclusão	51

CAPÍTULO 4

O método Hiper Autor..... 53

4.1 Justificativas	53
4.2 Objetivos do HiperAutor	54
4.3 O Ciclo de Vida	56
4.3.1 Modelo de Ciclo de Vida Clássico	57
4.3.2 O Modelo de ciclo de vida baseado em protótipos	59
4.3.3 O Modelo de Ciclo de Vida em Espiral.....	65
4.4 Método de desenvolvimento	68
4.4.1 Levantamento Inicial dos Requisitos.....	70
4.4.2 Elaboração do Esquema Geral do Conteúdo	75
4.4.3 Elaboração da rede de nós.....	78
4.4.4 Avaliação	81
4.4.5 Especificação detalhada do projeto	82
4.4.6 Implementação da versão 1	86
4.4.7 Avaliação	87
4.4.8 Novas Versões.....	87

CAPÍTULO 5

Conclusão..... 89

REFERÊNCIAS BIBLIOGRÁFICAS 92

A N E X O S

Anexo 1 - Formulários do Hiper Autor.....97

**Anexo 2- Exemplo de Utilização do Hiper
Autor.....102**

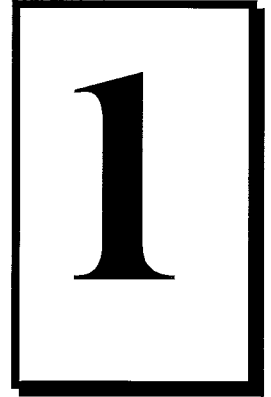
Lista de Figuras

◆ Figura 2.1	Histórico dos sistemas hipertexto/hipermídia nas últimas décadas.....	8
◆ Figura 2.2	Primeira página do capítulo 1 da aplicação Hands On! [Shne 91].....	18
◆ Figura 2.3	Index do aplicativo Hands On! [Shne 91]	19
◆ Figura 2.4	Mapa do Canadá, pertencente ao aplicação Guia das Oportunidades da Arqueologia Voluntária [Plais 91]	20
◆ Figura 2.5	Texto do artigo sobre o sítio arqueológico de Strathcona no Canadá, pertencente à aplicação Guia da Arqueologia Voluntária [Plais 91]	21
◆ Figura 2.6	Índice de artigos da aplicação Guia das Oportunidades da Arqueologia Voluntária [Plais 91].....	22
◆ Figura 2.7	- Classificação de Campbell e Goodman para sistemas de hipertexto	23
◆ Figura 2.8	Variedades de Nós [Mend 92]	24
◆ Figura 3.1	Espectro de tipos de mídia [Char 86]	30
◆ Figura 3.2	Retórica do g-IBIS	33
◆ Figura 3.3	Típica "frame" do KMS [Aksc 88]	35
◆ Figura 3.4	Tipos de Ligação	37
◆ Figura 3.5	"mundos " utilizados pelo método ThyDoc [Sobi 91]	39
◆ Figura 3.6	Conjunto de estruturas pré determinadas a serem preenchidas pelos autores de sistemas de hipertexto que estejam utilizando o ThyDoc.....	41
◆ Figura 3.7	Esquema do Software Nona Sinfonia de Beethoven [Garz 91]	45

◆ Figura 3.8 Qualidade de sistemas de Hipertexto [Mend 93].....	48
◆ Figura 3.9 Critérios utilizados para a avaliação da qualidade de aplicação do tipo hipertexto/ hipermídia	50
◆ Figura 4.1 O ciclo de vida Clássico [Ghe 91].....	58
◆ Figura 4.2 Ciclo de Vida baseado em prototipagem [Pre 92]	61
◆ Figura 4.3 O modelo de ciclo da vida de desenvolvimento de software utilizando a prototipagem descartável.....	62
◆ Figura 4.4 o modelo de ciclo da vida de desenvolvimento de software utilizando a prototipagem evolutiva	63
◆ Figura 4.5 o modelo de ciclo da vida de desenvolvimento de software utilizando a prototipagem operacional.....	64
◆ Figura 4.6 - O Modelo Espiral [Pres 92].....	66
◆ Figura 4.7 Atividades previstas no HiperAutor	69
◆ Figura 4.8 Ciclo de Vida do Ambiente de desenvolvimento de software do tipo hipertexto/hipermídia	70
◆ Figura 4.9- O processo de Análise do Problema [Leit 90]	71
◆ Figura 4.10 Formulário para Diagrama de Contexto	73
◆ Figura 4.11 Notação utilizada pelo Diagrama de Contexto	74
◆ Figura 4.12 Diagrama em blocos para a fase de Levantamento dos Requisitos	74
◆ Figura 4.13 Formulário Simplificado	76
◆ Figura 4.14 Diagrama em blocos para a atividade de Elaboração do Esquema Geral do Conteúdo	78
◆ Figura 4.15 Diagrama da Rede de Nós.....	79
◆ Figura 4.16 Notação da linguagem para a elaboração da Rede de Nós.....	80
◆ Figura 4.17 Diagrama em blocos da etapa de Elaboração da Rede de Nós.....	81
◆ Figura 4.18 Diagrama em blocos para a etapa de Avaliação	83

◆ Figura 4.19 Formulário detalhado	84
◆ Figura 4.20 Diagrama em blocos da etapa de Especificação Detalhada do Projeto.....	86
◆ Figura 4.21 Diagrama em blocos da fase de Implementação da Versão 1	87
◆ Figura 4.22 Diagrama em blocos da fase de Novas Versões.....	88

CAPÍTULO 1



Introdução

1.1 Justificativas

Os avanços tecnológicos da informática estão permitindo, cada vez mais, a difusão do uso de computadores em diferentes áreas. Interfaces mais amigáveis facilitam a manipulação de sistemas por usuários leigos e tem se tornado um fator chave para o aumento da produtividade e criatividade por parte dos mesmos. Planilhas eletrônicas, editores de texto, editores de imagens e, mais recentemente, sistemas de hipertexto/hipermídia, estão se popularizando entre profissionais de todas as áreas.

Atualmente, a hipermídia é a aplicação que vem ganhando mais destaque apesar de veicular um conceito bastante antigo. A idéia de se escrever um texto de maneira não linear, ou seja, se referenciando a anotações e outras fontes, já estava presente no Talmud e nos escritos aristotélicos sob a forma de comentários e referências a outras obras, constituindo protótipos de uma representação hipertextual [Conk 87].

Com o advento da computação, foi possível resgatar este conceito de hipertexto, uma vez que é possível fazer a integração de diferentes textos em um único ambiente. Mais recentemente, a partir dos avanços da indústria de hardware nas áreas de armazenamento de grandes quantidades de dados e tempo de acesso aos mesmos, foi possível estender este conceito para hipermídia, sistemas capazes de concatenar em uma única aplicação, diversos tipos de mídia, tais como imagens estáticas ou em movimento, som e texto [Shn 91]. Estas características os tornam ferramentas muito poderosas para a difusão de informações, o que implica em um grande número de possíveis empregos para esta tecnologia.

A nossa experiência na orientação da autoria e implementação de aplicações hipermídia voltadas para a área educacional demonstrou uma tendência para a formação de dois tipos distintos de equipe no que tange a elaboração de software deste tipo [Roch 92]. O primeiro tipo é formado pelos profissionais das áreas de conhecimento em que se deseja produzir as aplicações; estes, de maneira geral, possuem pouca ou nenhuma experiência em lidar com sistemas de computação. O segundo tipo de equipe, por sua vez, é constituído somente por profissionais da área de informática, analistas, programadores e engenheiros de software que apesar de não possuírem conhecimentos específicos da área de aplicação do software, contam com as habilidades necessárias para a elaboração de uma aplicação educacional.

Notou-se que as maiores dificuldades apresentadas por profissionais com pouca experiência em Informática relacionam-se basicamente a :

- ◆ definição e limitação do escopo e da base de dados a ser tratada no hipertexto;
- ◆ definição dos pontos de partida e chegada do documento;
- ◆ definição de uma técnica de trabalho, visto que os usuários ora usam uma abordagem "top-down", ora uma abordagem "bottom-up", o que prejudica a autoria e compromete a lógica do conteúdo;
- ◆ fragmentação da informação quebrando uma sequência lógica e não respeitando os objetivos do hiperdocumento [Sequ 92].

Verifica-se que este processo é artificial e às vezes arbitrário porque gera explosões (novos nós), que podem ser inconsistentes em relação à base de dados.

Como consequência dos fatores descritos acima, percebemos uma tendência em linearizar os documentos, como acontece na maioria dos livros textos, tornando os nós muito extensos com pouca presença de botões, o que representa uma contradição ao conceito de hipertexto de oferecer possibilidades para se ter um texto não sequencial e dinâmico.

Por outro lado, quando desenvolvemos aplicações hipermídia do tipo educacional a partir de equipes compostas somente por profissionais da área de informática percebemos as seguintes dificuldades:

- ◆ Inconsistência da base de dados, ocasionada pela falta de uma estratégia educacional válida para sua elaboração;
-

- ◆ Os objetivos pedagógicos planejados a priori, não foram alcançados ou foram somente parcialmente cumpridos;
- ◆ Consequente perda da qualidade global da aplicação.

Em todos os casos descritos acima, é evidente que não se conseguiu chegar a resultados plenamente satisfatórios pois, em nenhum deles, conseguimos gerar uma aplicação final que fizesse uso das vantagens técnicas que os sistemas hipermídia podem oferecer a aplicações do tipo hipertexto/hipermídia e, ao mesmo tempo, garantisse a consistência e validade do conteúdo didático formador da aplicação.

Perez [Pere 91] afirma que uma das maiores dificuldades relacionadas ao desenvolvimento de software do tipo hipermídia está ligada à formação da equipe de trabalho, devido a problemática inerente ao se tentar reunir profissionais de áreas diferentes em torno de um mesmo objetivo. Rocha [Roch 92] entende uma equipe interdisciplinar, que reúna profissionais de várias áreas, como " a solução adequada para o problema da autoria e implementação e a única possibilidade de se ter um produto de software educacional de boa qualidade ".

1.2 Objetivos da Tese

A partir das considerações anteriores propomos que o desenvolvimento de aplicações educacionais do tipo hipermídia, deve ser realizado por uma equipe multidisciplinar, que deverá conter

profissionais tanto da área técnica como especialistas da área de interesse da aplicação.

Nesta tese propõem-se um método para a especificação e projeto de aplicações hipermídia, o Hiper Autor. O objetivo deste método é apoiar uma equipe no processo de elaboração de uma aplicação hipermídia. Para tal, o HiperAutor se utiliza de técnicas de elicitação de requisitos e modelagem, de maneira a fornecer um conjunto de diretivas, organizando o pensamento e o trabalho da equipe ao longo do processo de desenvolvimento da aplicação.

1.3 Organização da Tese

Nos próximos capítulos apresentamos as razões que levaram à necessidade de elaboração de um método que apoiasse a construção de aplicações educacionais do tipo hipertexto/hipermídia e, apresentamos o método Hiper Autor.

O capítulo II apresenta conceitos básicos e a arquitetura de sistemas do tipo hipertexto/hipermídia, definindo a terminologia que será utilizada até o fim deste trabalho. Também são discutidas algumas estratégias de autoria, de modo a diferenciar os vários enfoques possíveis.

No capítulo III é apresentada uma revisão da literatura dos diversos métodos de autoria existentes. Fazemos uma divisão entre sistemas dependentes e independentes de ferramentas, apresentando um

estudo crítico de ambos. No final discutimos alguns critérios para a avaliação da qualidade de aplicações do tipo hipertexto/hipermídia.

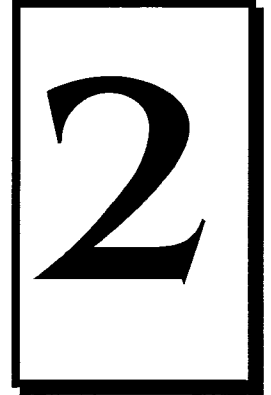
O capítulo IV apresenta um estudo para a determinação do ciclo de vida mais apropriado para o desenvolvimento de aplicações do tipo hipertexto/hipermídia. É apresentado o método Hiper Autor: o detalhamento das atividades previstas e os formulários para a documentação de produto.

O capítulo V corresponde à conclusão da tese. São apresentadas observações quanto a importância e aplicabilidade do trabalho e sugestões para futuras pesquisas.

O apêndice A apresenta um exemplo de aplicação do método Hiper Autor na elaboração da aplicação "Introdução ao computador com DOS para iniciantes".

CAPÍTULO 2

Hipertexto/ Hipermissão



2.1 Introdução

A possibilidade de organizar e recuperar informações de maneira não linear, através de recursos computacionais foi levantada após a Segunda Guerra Mundial por Vannevar Bush [Bush 45]. No seu estudo, Bush procura resgatar um conceito muito antigo, o da integração de diferentes textos em um único ambiente. A partir de então a evolução dos sistemas de hipermissão tem sido notável, como evidencia a figura 2.1.

Neste capítulo definiremos hipertexto/hipermissão, e apresentaremos conceitos básicos, discutindo algumas questões relativas à autoria, arquitetura e principais áreas de aplicação destes sistemas.

2.2 Definição

A literatura tem apresentado diferentes definições para o conceito de hipertexto. Selecionamos algumas definições que julgamos serem relevantes para uma melhor compreensão do que venha a ser hipertexto/hipermissão no contexto deste trabalho. Hipertexto pode ser definido como:

1945	Vannevar Bush publica o artigo 'Como devemos pensar' e propõe o Memex
1962	Douglas Engelbart publica o artigo 'Aumentando o Intelecto humano'
1965	Theodor Nelson inventa o termo HIPERTEXTO
1968	O primeiro sistema de hipertexto é criado na Universidade de Brown
1972	O sistema ZOG é desenvolvido na Universidade Carnegie Mellon
1983	Começa a segunda geração de sistemas de hipertexto
1985	É lançado o primeiro sistema de hipertexto comercial, GUIDE
1986	É lançado o NOTE CARDS da Xerox, que suporta figuras e animação
1987	É lançado o HIPERCARD e a versão PC do HIPERTIES
1990	Existem pelo menos vinte sistemas de hipertexto no mercado

Figura 2.1 Histórico dos sistemas hipertexto/hipermídia nas últimas décadas

◆ Uma forma de administração da informação, na qual os dados são armazenados em uma rede de nós. Os nós podem conter qualquer tipo de informação codificada (...) As ligações entre os nós podem ser ativadas por um dispositivo indicador (um mouse, por exemplo). A ativação de um nó que contém um texto faz com que este apareça "instantaneamente" na tela [Salg 92].

◆ Em um nível mais básico de uso, seria um sistema de gerenciamento de bancos de dados que permitiria ao usuário conectar telas de informações usando ligações associativas. Em nível mais sofisticado, poderia ser visto como um ambiente de software para trabalho cooperativo, comunicação e aquisição de conhecimento [Fide 88].

Para Slatin hipertexto pode ser considerado como uma coleção de textos, interligados eletronicamente através de nós, de modo a formar um sistema cuja existência é contingente ao computador [Slat 91]. Desta forma, o conceito de **hipertexto** está ligado a uma maneira de apresentar e recuperar informações, segundo um modelo de redes, compostas de nós interconectados por ligações. Encontramos na literatura a palavra hipermídia como sinônimo para hipertexto. Neste trabalho consideraremos hipermídia como sendo uma extensão do conceito de hipertexto, como definido abaixo:

" Hipertexto ou hiperdocumento são os termos utilizados por Theodor Nelson para definir o armazenamento e busca de informação não sequencial, mediados pelo computador. Uma extensão para este termo é hipermídia, que implica na interligação e navegação entre material armazenado em vários tipos de mídia: texto, figuras, som, música, vídeo, etc... " [Berk 91 b]

Os conceitos contidos nas definições selecionadas serão detalhados na próxima seção.

2.3 Conceitos Básicos de Hipertexto

Como já foi assinalado, podemos definir hipertexto como sendo um estilo de organização de dados sob a forma de uma rede de nós e elos. Passaremos a apresentar os conceitos básicos relacionados a este

tipo de sistema da forma em que se encontram sistematizados no livro *Hypertext/Hypermedia Handbook*, editado por Berk e Devlin 1991 [Berk 91a].

Nós

A unidade fundamental de informação no hipertexto é chamada de nó. De maneira idealizada seria uma pequena porção do documento que cobre um conceito. Podendo ser do tamanho de uma única tela ou de um livro inteiro, um nó pode conter texto, figuras ou outras formas de informação.

Ligações ou Elos

São ponteiros de um nó de um hipertexto para outro nó. Os elos representam a maneira que o autor do hipertexto relacionou os trechos de informação - nós - que compõem o conteúdo da aplicação.

Botão ou âncora

É a representação dos elos na tela. Pode ser uma palavra, frase ou até mesmo parte de uma figura. Quando ativado, ocorre um salto para outro nó do hiperdocumento. De um modo geral, os botões aparecem realçados, ou em cor diferente, no texto apresentado na tela. Não existe formato padrão para os botões.

Base de dados

É o conteúdo de informações contidas na aplicação. Neste estudo é sinônimo de escopo.

Sistema de autoria

É a ferramenta utilizada para implementar uma aplicação do tipo hipertexto/hipermídia. São ferramentas de autoria: o Author do Hyperties, o Toolbook, LinkWay e o NoteCards.

Metáfora

É a representação do conteúdo do hipertexto sob uma forma de representação do conhecimento de domínio do leitor. São metáforas comumente utilizadas enciclopédias e livros texto.

Navegação

Chamamos de navegação o processo de leitura de um hipertexto. Este processo tem início a partir do momento que o leitor aponta para qualquer um dos elos existentes a sua disposição. O elo selecionado, certamente, irá acessar algum outro nó onde ele, leitor, poderá novamente apontar para outro elo de sua escolha, e assim sucessivamente até que toda a base de dados tenha sido visitada.

Um dos resultados desta forma de leitura é a facilidade com que o leitor pode se perder em um emaranhado de nós. A este fenômeno chamamos de "perdido no hiperespaço" e se manifesta quando o leitor não consegue voltar a um determinado nó, ou se esquece a qual nó desejava se dirigir inicialmente. Uma outra consequência desta forma de leitura é o efeito "Serendipity*" que acontece todas as vezes que o leitor se desvia de seu objetivo principal para visitar

* Palavra inventada por Sir Horace Walpole no século XVIII, seu significado pode ser traduzido como:

"A faculdade de se fazer descobertas agradáveis e inesperadas por mero acidente".

outros nós, movido por algum interesse particular apenas. Este efeito não é particular aos sistemas de hipertexto, foi inicialmente detectado junto aos usuários de enciclopédias e dicionários em geral.

A desorientação pode ser minimizada através uma melhor estruturação da base de dados, o que permitirá o leitor uma orientação mais segura quanto ao acesso da informação desejada pelo mesmo. O segundo efeito não constitui exatamente uma desvantagem, uma vez que estimula a curiosidade e, portanto, aumenta o conhecimento geral daquele que estiver pesquisando a base de dados. É importante notar, que em ambos os casos é necessário que se empregue esforços no sentido de se melhorar a estruturação da base de dados de um sistema de hipertexto, o que só pode ser atingido através de metodologias consistentes para análise e projeto dos mesmos [Alty 93].

2.4 Autoria de sistemas do tipo hipertexto/hipermídia

Como texto não seqüencial, uma aplicação do tipo hipermídia pode permitir que o leitor crie diferentes caminhos de navegação. A originalidade deste tipo de sistema consiste no fato de que uma vez que a estrutura de conhecimentos de cada indivíduo é única, baseada em seu próprio conjunto de experiências e habilidades, as formas que ele prefere iniciar o trabalho, interagir e interrelacionar as informações seriam também diferentes [Jona 88]. Desta maneira, uma aplicação hipermídia é construída em parte pelo autor - que a criou -

e em parte pelo leitor - que toma as decisões sobre quais caminhos seguir. Leitores de hipermídia, então, assumem uma postura muito mais ativa do que aqueles de texto impresso [Shne 91].

Documentos em papel forçam o leitor a adotar uma forma linear de leitura - de cima para baixo e da esquerda para direita enquanto que a hipermídia encoraja o leitor a se mover de tópico em tópico (entenda-se de nó a nó) rapidamente e de maneira não sequencial.

Dada sua característica de possibilitar a organização e o acesso a um corpo de informações de forma não linear, a hipermídia permite, e até mesmo encoraja que o autor, ao escrever seu texto, guarde em nós, referências e outras informações textuais ou gráficas relacionadas diretamente ou não ao texto original. Também permite ao leitor tomar suas próprias decisões sobre quais ligações seguir e em que ordem, fazendo anotações no texto, a serem salvas separadamente do documento de referência. Conklin [Conk 87] entende que tais características dos hipertextos diminuiriam as restrições de pensar e escrever.

A literatura vem apontando a existência de grandes dificuldades na autoria e desenvolvimento de aplicações hipermídia.[Alty 93], [Maye 93] [Roch 92], [Shne 91], [Niel 90] [Garz 90]. Estes problemas decorrem do fato de que o documento em hipermídia oferece, a cada nó, diversas opções para o leitor, que toma as decisões sobre que caminhos seguir. Desta forma, o autor é responsável por fazer uma previsão de todos os caminhos possíveis de serem visitados pelo leitor

e, também de organizá-los de modo a obter consistência no conteúdo das informações.

As técnicas sofisticadas de aquisição e organização de informações apresentadas pelos computadores, como é o caso de hipertexto/hipermídia, podem a princípio parecer confusas para o usuário leigo, prejudicando seu processo de aprendizagem.

Para minimizar esta problemática, os sistemas de hipertexto/hipermídia podem se basear em metáforas, já conhecidas pelos usuários, para facilitar a compreensão. Wurman define as metáforas da seguinte forma:

"No início, a idéia em termos de apresentação eletrônica de dados era imitar projetos já existentes de apresentação de dados de modo que as pessoas não ficassem tão confusas. Livros de referência estão entre as fontes de mais fácil compreensão pois as pessoas foram criadas com eles, e sabem usá-los pois foram laboriosamente treinados durante a escola... O único problema é que além da escola, não possuímos nenhum treinamento quanto a maneira de acessar qualquer tipo de dados. " [Wurm 89]

Autores de aplicações do tipo hipertexto/hipermídia se utilizam de metáforas de maneira a estabelecer convenções que os leitores possam compreender e, desta forma, superar qualquer dificuldade que possa ser encontrada na interface usada. Dentre as várias metáforas existentes para a confecção de sistemas de hipertexto/hipermídia, escolhemos duas para dissertação nesta obra, por serem

as mais utilizadas nos aplicativos educacionais. São elas, as metáforas de livro eletrônico e de excursão.

2.4.1 A metáfora de Livro Eletrônico

Neste tipo de organização, a informação é dividida em parcelas do tamanho de parágrafos ou páginas. Esta informação é alocada a páginas ou artigos, interrelacionados através de elos. O conjunto de páginas formará o corpo total da informação, ou livro.

Para apoiar a construção de aplicativos baseados neste tipo de metáfora, existem algumas estruturas especiais que analizaremos a seguir:

- ◆ Capítulos - divisões em assuntos da base de dados
 - ◆ Índice - lista todos as páginas ou artigos disponíveis na aplicação
 - ◆ Elos- ligações entre uma palavra ou expressão (hotword) existente em uma determinada página e outra página.
 - ◆ Notas- podem ser globais ou de margem. A primeira é geral para todas as páginas que compõem o aplicativo e o segundo é particular a uma determinada página. [Maye 93]
 - ◆ Histórico - mecanismo que permite ao usuário se realocar a qualquer página que o mesmo já tenha visitado. Também relaciona os nomes de todas as páginas que já foram vistas
 - ◆ Marcadores - permitem o usuário a fazer uma marcação específica em determinadas páginas. Em um momento posterior é possível listar estas páginas para um novo acesso.
-

- ◆ "Migalhas de Pão " (Bread Crumbs) - permitem que o usuário faça a marcação das páginas que percorreu. É uma alusão ao conto de fadas de João e Maria, onde as duas crianças marcaram sua trilha na floresta através de migalhas de pão. [Berk 91]
- ◆ Busca - permite a busca de uma sequência de caracteres pela base de dados

Existem uma série de vantagens na adoção da metáfora de livro eletrônico. Apontamos como a maior delas, o fato de todos os usuários serem familiarizados com a organização visual e navegacional fornecidas por qualquer livro texto [Shne 91]. A adoção das convenções existentes em um livro, garante a maioria dos usuários a capacidade de utilizar o aplicativo com um mínimo de instruções .

2.4.2 A Metáfora da Excursão

Textos lineares fornecem estruturas melhores de se garantir que os leitores vejam todos os pontos importantes do conteúdo do que os sistemas de hipertexto. Este fato é devido à forma ordenada e sequencial que é imposta a seus leitores. Os sistemas de hipertexto/hipermídia por sua vez, oferecem uma flexibilidade muitas vezes indesejada, uma vez que não podem garantir que os usuários acessem determinadas informações, as vezes fundamentais à compreensão do conteúdo global do aplicativo.

A utilização da metáfora de excursão tenta minimizar esta problemática provendo os autores de sistemas de hipertexto/hipermídia com uma maneira de especificar caminhos ordenados dentro da base de dados. Desta forma evitamos que os

leitores se percam, na medida em que garantimos que estes leiam os nós em uma ordem pré determinada [Shne 91] .

Esta é a metáfora que será adotada por este trabalho, uma vez que permite que os autores, à medida em que vão definindo os caminhos obrigatórios dentro da base de dados de sua aplicação, possam, ao longo do processo, embutir sua estratégia de ensino.

A seguir, apresentaremos dois exemplos de aplicativos desenvolvidos a partir das metáforas apresentadas.

2.4.3 Exemplo de uma aplicação construída a partir da metáfora de livro eletrônico: O Hipertexto Hands On!

Este é um guia para a construção de hipertextos que foi publicado tanto em papel quanto na forma de livro eletrônico, de maneira a permitir que seus leitores pudessem fazer uma comparação entre as duas formas de mídia[Shne 89]. Na edição eletrônica o acesso à informação foi feito da seguinte maneira: via uma plataforma hipertextual que permite o ingresso em qualquer um dos capítulos e, via o index, que remete o usuário a qualquer um dos artigos que este selecionar.

Esta plataforma hipertextual fornece pontos de partida pelos quais os leitores podem ir se aprofundando. A medida em qualquer um dos

capítulos for selecionado, o leitor recebe um pequeno sumário do conteúdo deste e uma lista dos artigos contidos no mesmo. A todo momento é permitido ao leitor retornar a plataforma. Este artifício diminuiu o efeito "Perdido no hiperespaço", o que pode ser resumido como a perda de direção por parte do leitor.

As figuras 2.2 e 2.3 mostram respectivamente o índice e a primeira página do capítulo 1 do aplicativo Hands On! .

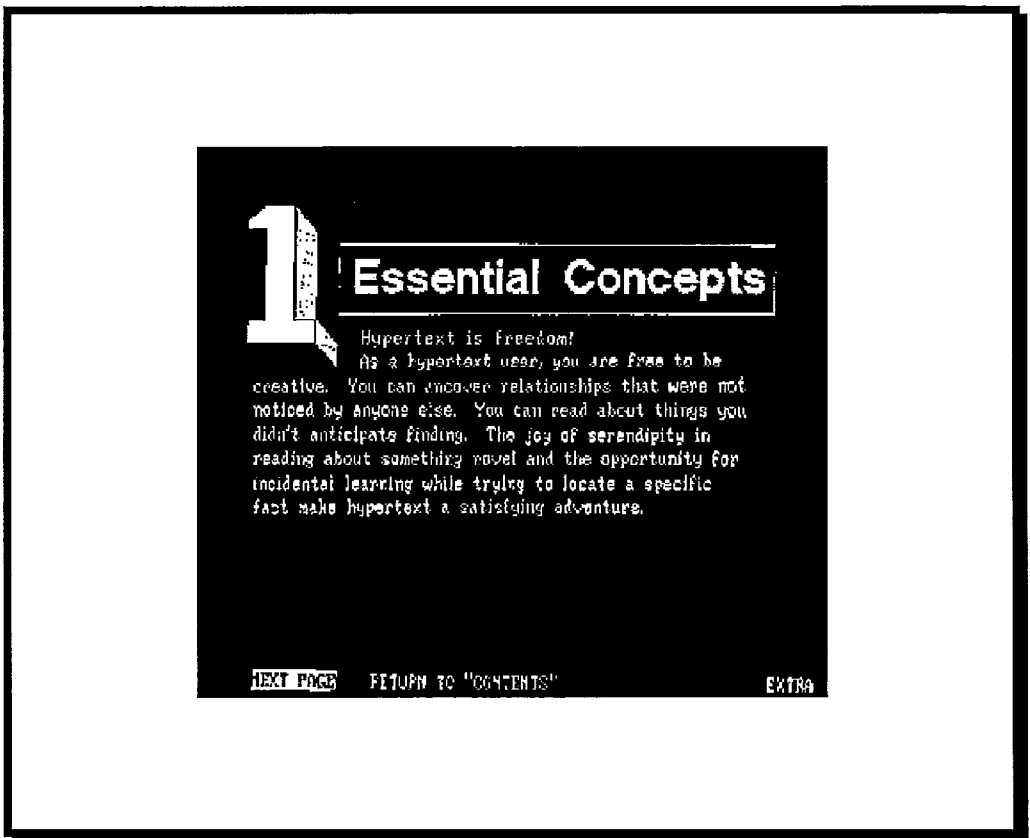


Figura 2.2 Primeira página do capítulo 1 da aplicação Hands On! [Shne 91]

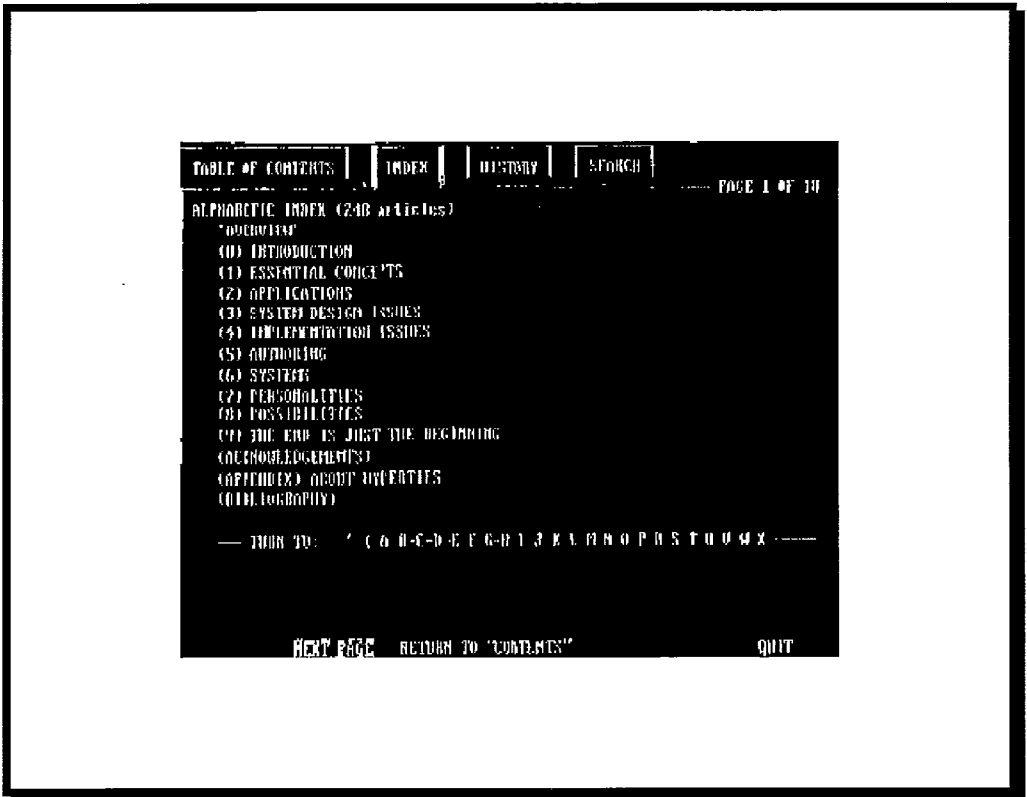


Figura 2.3 Index do aplicativo Hands On! [Shne 91]

2.4.4 Exemplo de um aplicativo construído a partir da metáfora de excursão: Guia das Oportunidades da Arqueologia Voluntária

Este aplicativo foi desenvolvido pelos alunos e professores do departamento de História da Universidade de Maryland através da colaboração do Instituto de Interação Homem-Computador, da mesma universidade.

O aplicativo foi desenvolvido para uma exibição em vários museus e é constituída de mais de 200 artigos. O conteúdo inclui informações sobre escavações arqueológicas em todo o mundo, descrições dos períodos históricos e algumas sugestões para aqueles que desejem se tornar arqueólogos.

Neste aplicativo é fornecido aos usuários uma excursão a cada um dos sítios arqueológicos disponíveis. A seleção deste pode ser feita através dos mapas ou listagens das opções. Também é fornecida a opção de listagens de todos os artigos que compõem a base de dados, para que o leitor possa ter acesso imediato aos mesmos. As figuras 2.4, 2.5 e 2.6 mostram respectivamente um dos mapas, o texto do artigo sobre o sítio arqueológico de Strathcona no Canadá e o índice de artigos do aplicativo.

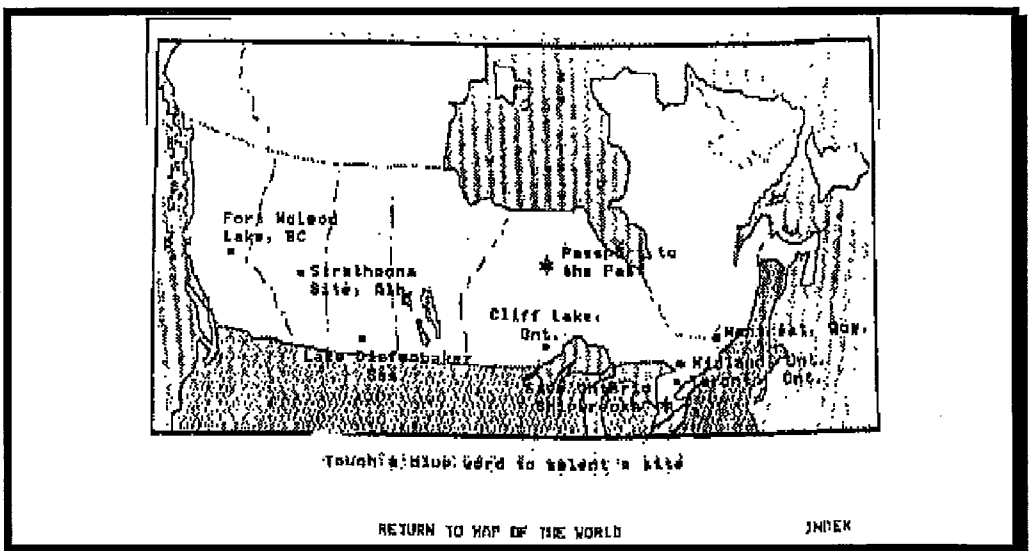


Figura 2.4 Mapa do Canadá, pertencente ao aplicação Guia das Oportunidades da

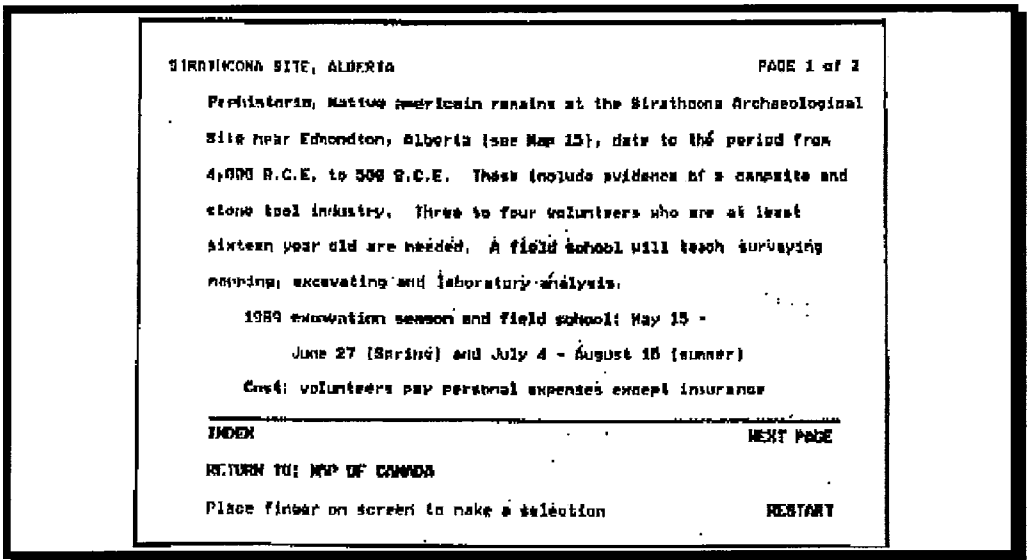


Figura 2.5 Texto do artigo sobre o sítio arqueológico de Strathcona no Canadá, pertencente à aplicação Guia da Arqueologia Voluntária [Plais 91]

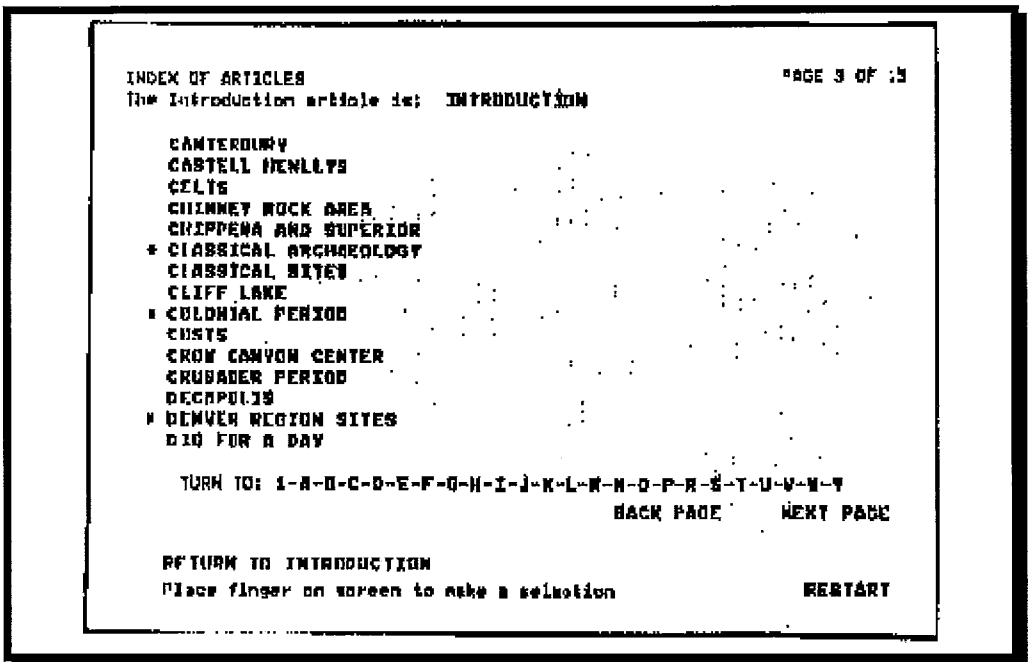


Figura 2.6 Índice de artigos da aplicação Guia das Oportunidades da Arqueologia Voluntária [Plais 91]

2.5 Arquitetura Básica dos sistemas de Hipertexto

Independente do tipo de implementação a ser utilizado, os sistemas do tipo hipertexto/hipermídia mantêm uma uniformidade quanto a sua arquitetura básica. Segundo a classificação de Campbell e Goodman, os sistemas de hipertexto dividem-se em três níveis, como ilustra a figura 2.7:

- ◆ Nível de apresentação: interface com o usuário
- ◆ Nível de máquina abstrata: nós e elos
- ◆ Nível de banco de dados: armazenagem, compartilhamento de informações e acesso à rede. [Camp 88]

No nível de banco de dados são realizadas as atividades relacionadas à armazenagem de informações. São elas: a divisão da informação nos vários dispositivos de armazenamento de dados, tais como discos rígidos e óticos, mecanismos que tornem a busca de dados rápida e eficiente, permitir o acesso de múltiplos usuários e segurança do sistema.

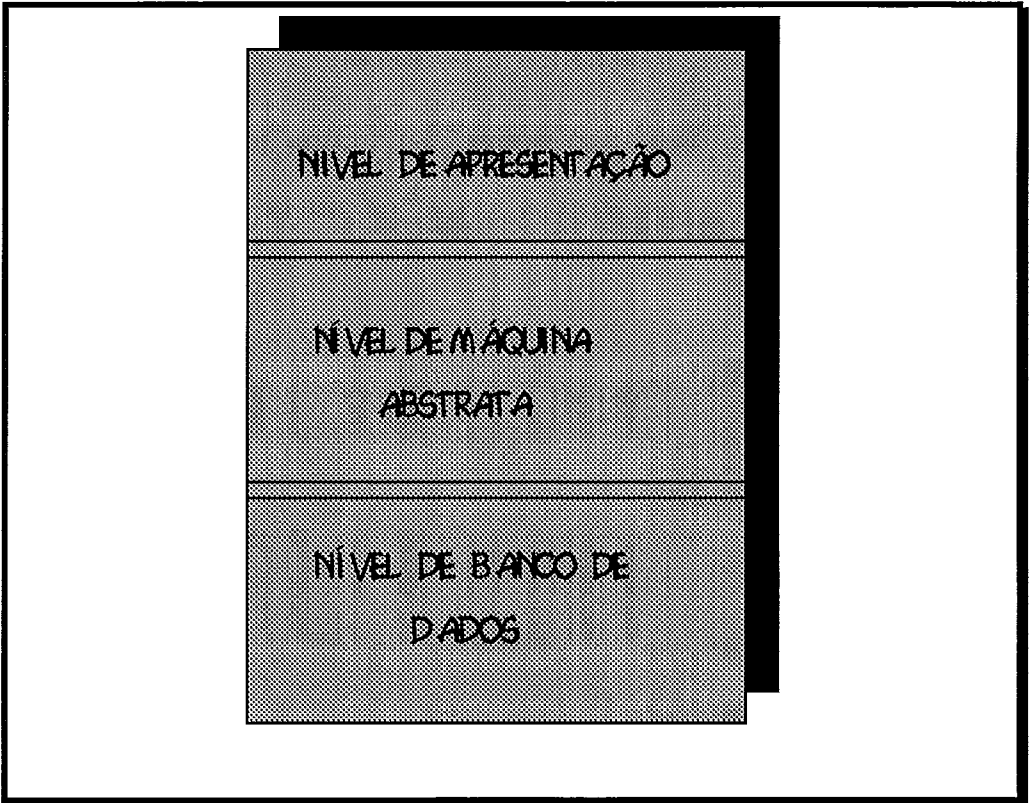


Figura 2.7 - Classificação de Campbell e Goodman para hipertexto [Niel 90]

Neste nível, os elementos básicos de hipertexto, nós e elos, não possuem significado particular, sendo considerados como objetos. Nielsen afirma que este nível lida com questões relacionadas à especificação de banco de dados, que não tem relação ao sistema de hipertexto [Niel 90].

O nível de interface com o usuário é responsável pela maneira como a informação será mostrada ao usuário final. Inclui os comandos que serão disponíveis, a maneira com que os nós e elos serão demonstrados na tela e se algum tipo de diagrama, do tipo histórico dos nós visitados, estará presente.

É no nível intermediário, o da máquina abstrata, que serão determinados a natureza dos nós e elos que farão parte da aplicação, seus atributos e ligações. Mendes [Mend 92] enumera quatro

variedades para os nós disponíveis em um hiperdocumento: nós compostos, nós tipados, nós não tipados e nós semi-estruturados, como mostra a figura 2.8

Nós tipados - São nós destinados a realização de tarefas diferentes do ordinário, tais como realizar anotações ou comentários. Estes nós se distinguem dos demais por possuírem atributos, chamados tipos, que são capazes de realizar diferentes operações, dependendo da natureza do nó.

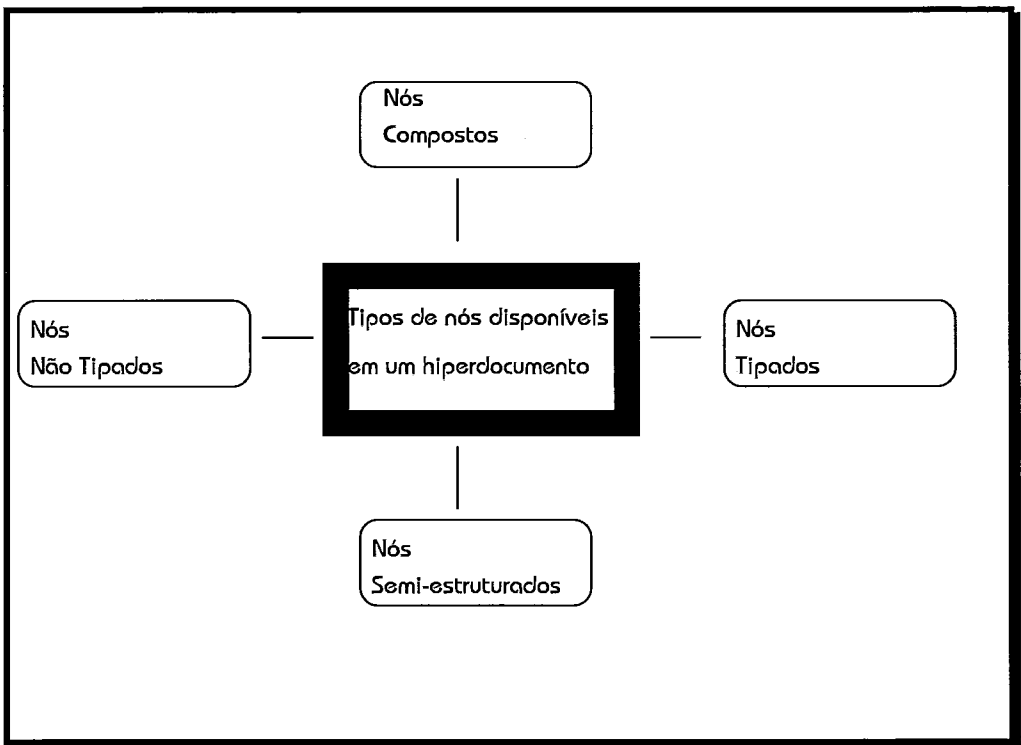


Figura 2.8 Variedades de Nós [Mend 92]

Nós não tipados - Representam qualquer tipo de informação, textual, gráfica ou sonora. Possuem um número determinado de operações que podem realizar, não havendo nenhum tipo de discriminação entre os nós. Desta forma, as operações que podem ser efetuadas sobre

quaisquer um dos nós são sempre as mesmas. Atributos comuns aos nós são nome, autor e data de criação.

Nós Compostos - Chamamos nó composto o grupamento de um conjunto de nós que possuem uma mesma característica. Estes nós são organizados de maneira que qualquer operação executada sobre um componente terá o mesmo resultado em todos os outros. Estes nós podem ter atributos associados.

Nós Semi-Estruturados - Este tipo de nós possui dois tipos de informação: não estruturada e estruturada. Definimos como informação não estruturada os gráficos, imagens estáticas e dinâmicas, sons e textos que compõem a base de dados do hipertexto. Informação estruturada diz respeito a campos específicos existentes nos hiperdocumentos, criados através de uma linguagem específica - Script.

Para mais informações sobre tipos de nós e suas características, referenciar-se [Mend 92].

O nível de máquina abstrata é o que apresenta maior campo para pesquisa futura em direção a uma padronização dos sistemas de hipertexto/ hipermídia, uma vez que o nível de banco de dados é dependente do tipo de equipamento a ser usado, e o nível de interface, por sua vez, varia de um sistema de hipertexto para outro.

Neste trabalho vamos nos concentrar nos dois últimos níveis, por apresentarem questões relevantes à autoria de sistemas hipertexto/hipermídia.

2.6 Conclusões

Os sistemas do tipo hipertexto/hipermídia têm se mostrado versáteis para a confecção de aplicações em diferentes áreas do conhecimento. Segundo Ben Shneiderman [Shne 91], todas as bases de dados que se enquadram nas seguintes regras são elegíveis para se tornarem aplicações do tipo hipertexto/hipermídia:

- ♦ constituírem um corpo de informação que possa ser organizado em vários trechos.
- ♦ estes trechos estarem interrelacionados.
- ♦ o usuário ser capaz de compreender cada fração da informação isoladamente.

Em vários campos já existem aplicações disponíveis. Nielsen destaca cinco grandes áreas onde o desenvolvimento de hipertexto/hipermídia vem sendo mais intenso: Informática, Negócios, Educação, Aplicações "Intelectuais" e Entretenimento [Niel 90]. A figura 2.9 mostra esta divisão, apresentado sub-áreas que cada uma pode conter.

Sistemas de hipertexto/hipermídia estão se tornando usuais para diferentes domínios de aplicação e sob diferentes plataformas - desde micro computadores PC compatíveis até estações de trabalho. Alty, em recente artigo lança a pergunta "Multimídia: Nós temos a tecnologia mas temos a metodologia?" e faz uma previsão de que haverá uma forte aceleração no desenvolvimento de aplicações bem sucedidas desde que algumas destas dificuldades sejam superadas. Tais dificuldades relacionam-se basicamente à falta de uma metodologia eficiente de apoio à autoria de sistemas do tipo hipertexto/hipermídia [Alty 93].

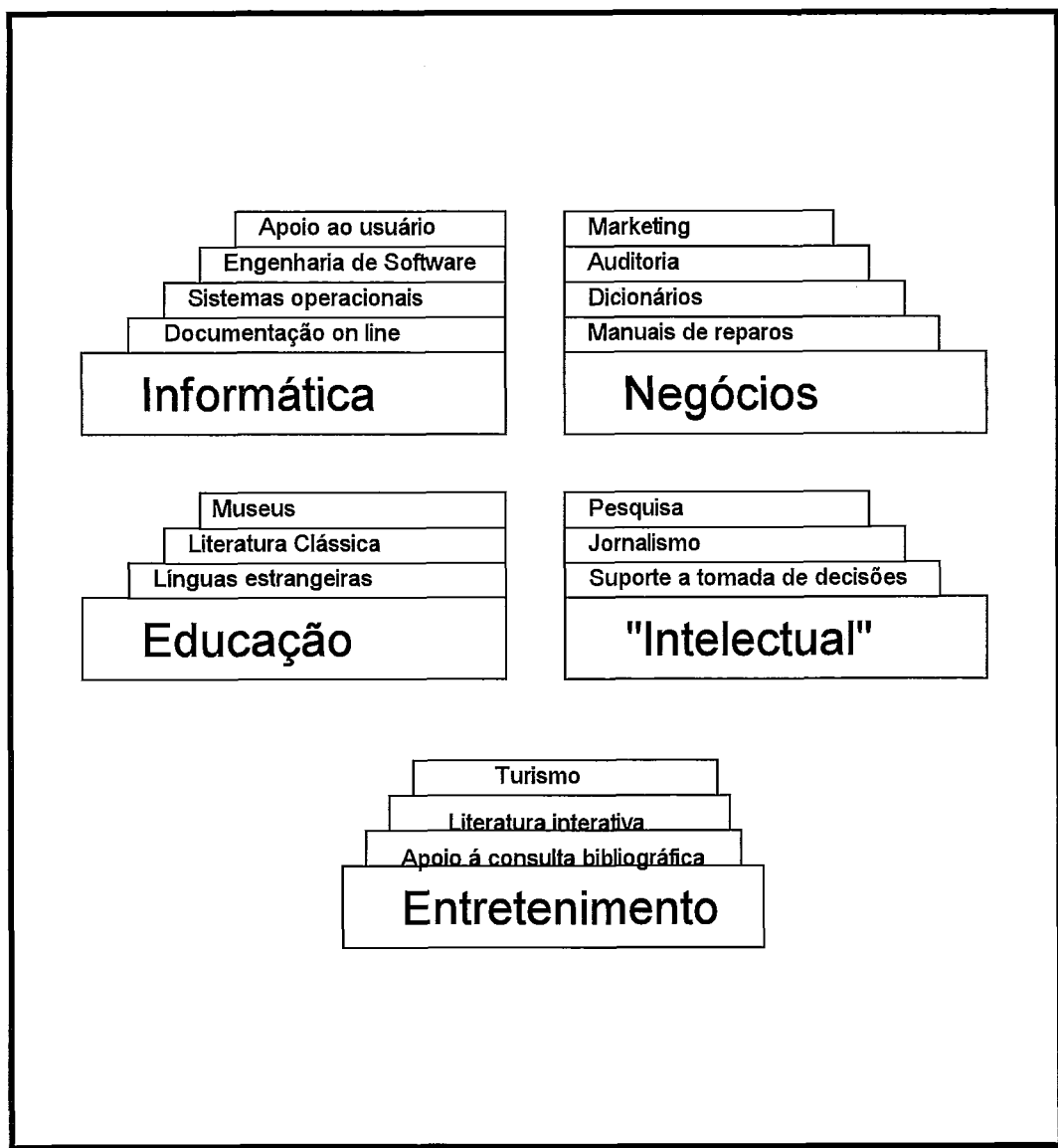


Figura 2.9 Áreas de utilização de hipertexto/hipermídia.[Niel 9o]

É no sentido de minimizar estas dificuldades que concebemos o presente trabalho. Nos capítulos subseqüentes revisamos a literatura sobre as metodologias para desenvolvimento de sistemas do tipo hipertexto/hipermídia e apresentamos nossa proposta **HIPER AUTOR**.

O desenvolvimento de aplicações hipermídia



3.1 Introdução

As últimas décadas têm mostrado uma grande evolução no que se refere a tecnologia relacionada ao processo de produção de software. Entretanto, a disponibilidade de novos recursos de hardware, associada à diminuição de seus custos são fatores de grande impacto na área de desenvolvimento de aplicações, pois tem levado a um aumento substancial na demanda por novos produtos e na complexidade destes. No que se refere ao desenvolvimento de aplicações hipermídia, as barreiras tecnológicas encontradas pelos primeiros desenvolvedores, tais como sincronização de imagens em movimento, vem gradativamente sendo vencidas, provando que não existem razões do ponto de vista tecnológico, que impeçam o desenvolvimento de sistemas hipermídia. James Alt [Alt 93], em recente congresso, afirma que as dificuldades relacionadas ao desenvolvimento de sistemas hipertexto/hipermídia não estão

relacionadas à falta de tecnologia e sim à necessidade da obtenção de uma metodologia eficaz para o desenvolvimento de seus aplicativos.

Fica bastante clara, à medida em que aumentam a complexidade e sofisticação destes sistemas, a necessidade da adoção de um enfoque metodológico, que seja capaz de apoiar a equipe durante o desenvolvimento de uma aplicação em hipertexto/hipermídia. A quantidade de informações envolvidas neste processo é muito grande e demanda a utilização de algum tipo de método que discipline e guie o processo de desenvolvimento.

Rocha define um método como um conjunto de diretivas para a seleção e aplicação sistemática de técnicas e instrumentos, de forma a organizar o pensamento e o trabalho do usuário ao longo do processo de desenvolvimento de software [Roch 87].

Métodos são compostos de conceitos ou "leis fundamentais", chamados princípios, que definem todo tipo de ação que são capazes de realizar. Segundo Ross [Ross 75] são sete os princípios básicos que compõem um método: modularidade (estruturação apropriada do produto), abstração (identificação de propriedades comuns), ocultamento (capacidade de tornar inacessível informações desnecessárias), localização (capacidade de agrupar elementos semelhantes), uniformidade (utilização dos mesmos formatos para descrever a informação), completeza (presença de toda a informação) e capacidade de confirmação (descrição da informação necessária para descrever o problema). A utilização destes princípios resulta em

uma série de vantagens, tais como auxílio à compreensão, redução da complexidade, limitação do impacto de mudanças e precisão.

Um aspecto bastante importante em métodos é o tipo de mídia que será utilizado para implementar os princípios mencionados. Charette, [Char 86] faz uma representação gráfica, figura 3.1, onde ilustra o espectro de possibilidades desde a menos formal até mais formal das representações. Cada um dos tipos de mídia utilizado difere dos outros pelo nível de interpretação necessário à sua compreensão e, pelo tipo de informação que contém. O tipo de representação selecionado é fundamental para determinar a habilidade do método em capturar e fornecer dados sobre o domínio de sua aplicação. Desta forma, o tipo de mídia utilizado é um fator determinante da utilidade do método para uma aplicação específica [Char 86].

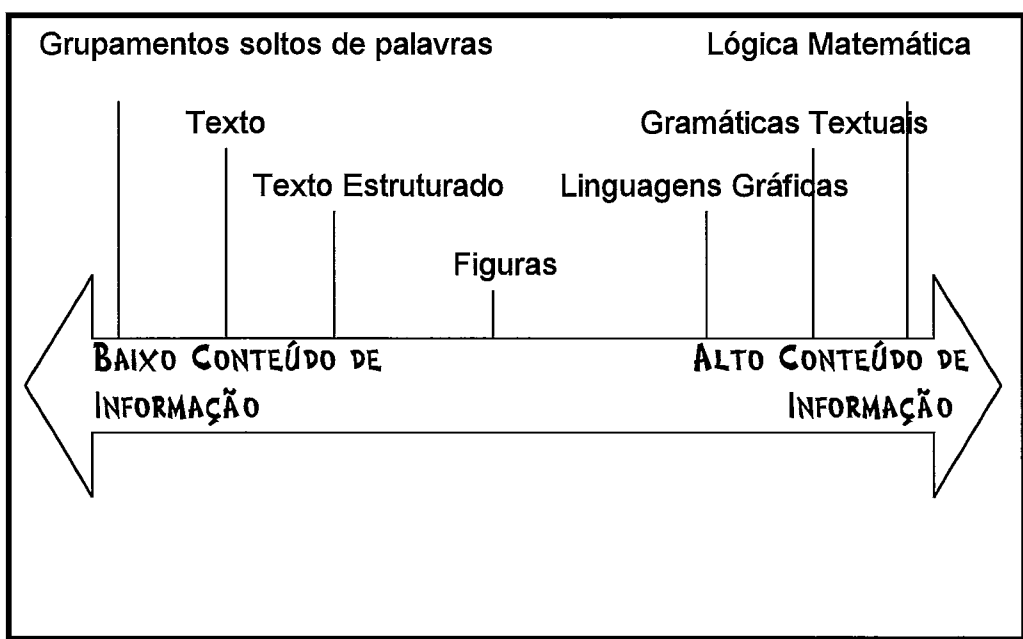


Figura 3.1 Espectro de tipos de mídia [Char 86]

A seleção de um método, portanto, não deve ser realizada sem levar em conta os fatores abordados anteriormente. Devemos sempre ter em mente que o método a ser escolhido deve facilitar a comunicação entre clientes, usuários e desenvolvedores ao mesmo tempo em que auxilia na compreensão e decomposição de problemas. Somente desta forma será possível a construção de produtos de qualidade.

Nas seções seguintes descrevemos os métodos atualmente disponíveis que visam apoiar o desenvolvimento de aplicações hipermídia.

3.2 Métodos de autoria para sistemas hipermídia

Um estudo da literatura permite, de imediato, estabelecer duas classes de métodos para a autoria de sistemas do tipo hipermídia. Na primeira tem-se métodos para a autoria, embutidos em alguma ferramenta para a construção dos sistemas. É o caso do g-IBIS [Conk 88], KMS [Aksc 88], e HELENA [Petr 93]. Uma outra classe contém métodos cuja existência não é contingente a nenhum software existente, ou seja, possibilitam a análise e projeto da aplicação independente da ferramenta que será utilizada para sua implementação. É importante, também se considerar a extrema novidade do tema, o que leva a métodos propostos muito recentemente e ainda sem o desejável nível de maturidade, fruto de sua aplicação em diferentes projetos.

3.2.1 Métodos de autoria dependentes de ferramentas para a implementação

3.2.1.1 g-IBIS [Conk 87]

g-IBIS é um sistema de hipertexto projetado para facilitar a coleta de deliberações ("design deliberations") na fase inicial de projeto. Esta ferramenta é a implementação do método de autoria chamado IBIS (Issue Based Information Systems) desenvolvido por Horst Rittel e é baseada no princípio de que o processo de análise de problemas complexos é realizado através de conversações entre membros de uma equipe composta por projetistas, clientes e implementadores. Neste processo, cada participante da equipe contribui com conhecimento e pontos de vista relacionados à solução dos *tópicos* abordados pelo projeto. O modelo g-IBIS é baseado nas articulações entre os *tópicos* que representam pontos chave para o projeto.

Cada um dos *tópicos* pode conter vários *posicionamentos*, que são definidos como afirmações que resolvem as questões relativas a um determinado *tópico*. Um *posicionamento*, por sua vez, pode conter um ou mais *argumentos* que o apoiam ou se opõem a ele. A figura 3.2 mostra a retórica prevista pelo g-IBIS.

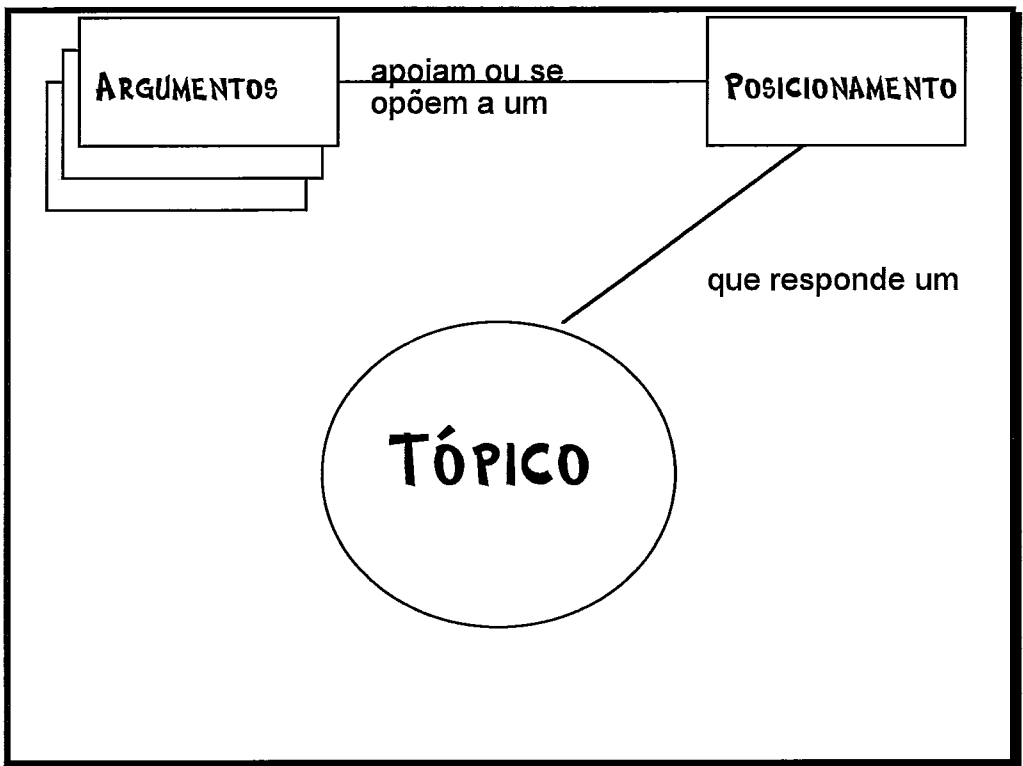


Figura 3.2 Retórica do g-IBIS [Conk 87]

O processo de autoria se inicia no momento em que um dos participantes coloca em discussão um tópico. O resto da equipe deve, então, apresentar posicionamentos que resolvam a problemática sugerida pelo tópico em questão. A seguir, são colocados argumentos que podem apoiar ou se opor aos posicionamentos em pauta. O conjunto de todas as possibilidades abordadas representa a modelagem do sistema.

Este método consegue, através de estruturas pré-determinadas, modelar a semântica do domínio de aplicação [Garz 90]. Esta característica o torna bastante poderoso para o desenvolvimento de aplicações que se enquadram neste domínio - voltadas para a

documentação de deliberações de projeto - mas, é pouco útil para a modelagem de aplicações mais genéricas.

3.2.1.2 KMS [Aksc 88]

KMS (Knowledge Based System) é uma ferramenta para a construção de aplicações hipermídia de larga escala, voltados para o trabalho cooperativo. O método de autoria implícito nesta ferramenta supõe que a produtividade da equipe desenvolvedora da aplicação pode ser aumentada através da utilização de um modelo de dados simplificado. Neste modelo é definida uma única estrutura, chamada "frame" ou nó, para o armazenamento das informações que constituem o conteúdo da aplicação. Cada nó possui um nome, título, ligações que remetem a outros nós e áreas pré-estabelecidas para a elaboração de seu conteúdo e anotações. A figura 3.3 exemplifica um típico nó ou "frame" do KMS.

Os nós estão organizados em estruturas hierárquicas de maneira a incentivar uma autoria do tipo "top down" [Garz 90]. Cada um dos nós pertence a uma única hierarquia e, suas ligações, remetem a outros nós localizados em um patamar abaixo. O processo de autoria consiste na fragmentação da base de dados de modo a encaixá-la neste tipo de estrutura. Este tipo de modelagem vem sofrendo duras críticas [Berk 91] e [Schn 91], pela pouca mobilidade e fracos recursos de representação da informação oferecidos.

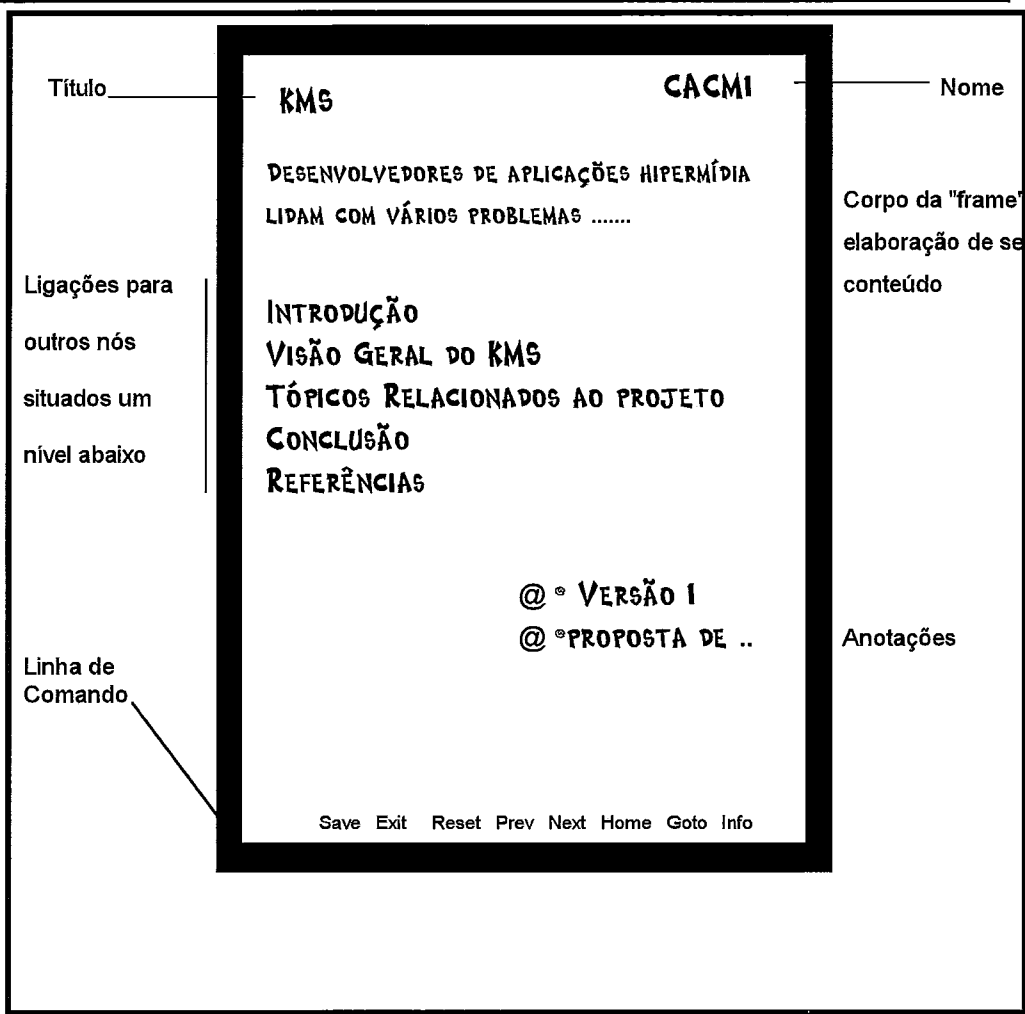


Figura 3.3 Típica "frame" do KMS [Aksç 88]

3.2.1.3 HELENA [Petr 93]

HELENA (Hypermedia ELetronic ENcyclopedia Authoring tool) é uma ferramenta criada para auxiliar a construção de aplicações hipermídia voltadas para ambientes de enciclopédias eletrônicas de uso geral.

A ferramenta é dividida em três programas: *administrator*, *executor* e *constructor*. O *administrator* é voltado para os usuários de enciclopédias eletrônicas, realizando o registro e o processamento de dados estatísticos resultantes da interação com o sistema. O *executor* é

voltado ao usuário final do sistema e, o *constructor* fornece as ferramentas necessárias à autoria.

O método de autoria suposto por HELENA é baseado na metodologia tradicional de criação de manuais. Desta forma o processo de autoria embutido na ferramenta consiste na execução dos seguintes estágios:

- ◆ 1o. Estágio - Construção de um glossário sobre o domínio da aplicação;
- ◆ 2o. Estágio - Construção de um glossário alfabético;
- ◆ 3o. Estágio - Preparação de artigos e figuras sobre cada um dos termos abordados;
- ◆ 4o. Estágio - Determinação de ligações entre os termos.

As ligações entre os termos podem ser de dois tipos: associativas e estruturais. As ligações associativas relacionam dois termos que se referem a um mesmo tópico, enquanto que as ligações estruturais se relacionam ao modo de organização da informação. HELENA organiza os termos em estruturas de árvore, onde os conceitos localizados em níveis inferiores representam uma especialização ou parte de um conceito em um nível superior. A figura 3.4 mostra os possíveis tipos de ligação entre os termos [Petr 93].



Figura 3.4 Tipos de Ligação

3.2.2 Métodos de autoria independentes de ferramentas

3.2.2.1 As diretrizes propostas por Ernest Perez

Na tentativa de tornar a autoria de sistemas hipertexto/hipermídia um processo mais metódico, Ernest Perez [Pere 91] propõe uma série de diretrizes para a elaboração de uma aplicação deste tipo. Estas diretrizes estão organizadas em quatro etapas que formam um plano geral para a criação de hiperdocumentos em larga escala. Neste trabalho vamos enumerá-las de maneira sucinta, como se segue:

- ◆ 1a. Etapa: Planejamento. Esta etapa consiste em extensivo planejamento estratégico, definição de objetivos, análise, especificação e estudo de custos.

◆ **2a. Etapa: Coleta de dados.** O conteúdo da base de dados de um hiperdocumento pode se originar de dois tipos de fonte. O primeiro é a conversão de arquivos texto ou banco de dados existentes e armazenados na forma de arquivo digital. Um segundo tipo é a elaboração ou digitalização de material especialmente para a confecção do hiperdocumento. Nesta etapas serão realizadas ambas funções.

◆ **3a Etapa: Organização da informação.** Para a realização desta etapa, o autor recomenda a utilização de banco de dados para realizar a tarefa de coleta e organização de grandes quantidades de dados. A utilização de Banco de Dados facilitará a visualização do conteúdo dos nós, além de apoiar a busca de palavras ou frases dentro da base de dados. Alguns sistemas deste tipo sugeridos pelo autor são o Ask/Sam, Nutshell Plus e Cuadra Star.

◆ **4a Etapa: Preparação final.** Nesta fase é feita a formatação da informação de modo a adequá-la ao formato da ferramenta de autoria que será utilizada para a implementação do sistema. É importante frisar que até este ponto do desenvolvimento, as diretivas propostas servem como um método de modelagem conceitual independente do software de autoria.

Apesar de não fazer qualquer referência a critérios para a avaliação da qualidade da aplicação, o autor prevê a possibilidade de realização de uma inspeção, feita pelos autores, com o objetivo de avaliar o sistema. O conjunto de diretivas apresentado por Perez, apesar de bastante conciso, mostra claramente a necessidade de uma metodologia de

autoria para aplicações hipermídia. Esta deve ser baseada em conceitos de organização da informação e ferramentas que apoiem o acesso à base de dados, de maneira a facilitar o processo de desenvolvimento de aplicações do tipo hipertexto/hipermídia.

3.2.2.2 O método ThyDoc [Sobi 91]

ThyDoc é um método para a modelagem de documentação "on-line" sob a forma de hipertexto. Este método é fundamentado na Engenharia de Software Baseada em Conhecimento, através da utilização do conceito de divisão da base de dados em vários domínios de aplicação. Estes são chamados "mundos" da aplicação e representam o conteúdo, forma e a apresentação da informação que compõe o hiperdocumento.

Thy Doc modela a base de dados de um hiperdocumento representando os três diferentes tipos de conhecimento - conteúdo, forma e apresentação - através da utilização dos seguintes domínios: mundo do assunto, mundo de documentação e mundo da apresentação. A figura 3.5 mostra esta divisão.

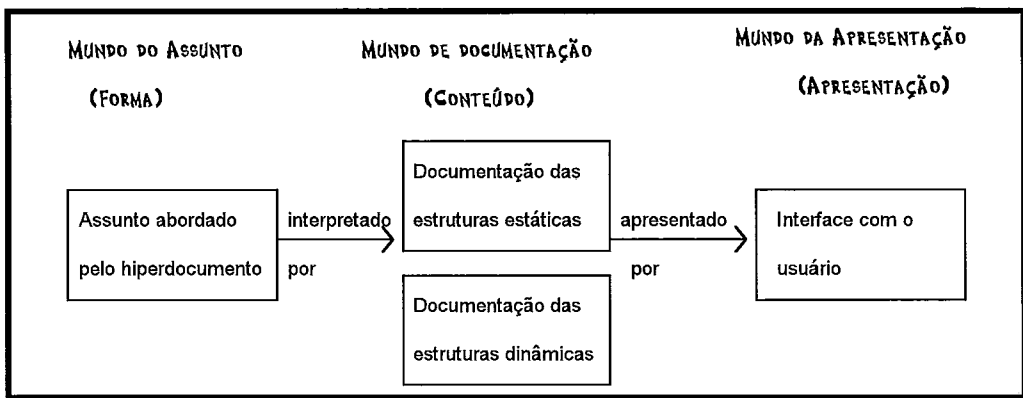


Figura 3.5 "mundos" utilizados pelo método ThyDoc [Sobi 91]

Estes mundos englobam os seguintes tipos de conhecimento:

- ♦ **Mundo do assunto** - Reúne o conhecimento sobre o assunto que é o sujeito da aplicação. Neste domínio, o conhecimento que é descrito de maneira informal no hiperdocumento - através de figuras, gráficos e comentários- será modelado mais formalmente através de estratégias semelhantes às usadas para a modelagem da especificação de requisitos de software, tais como o modelo de Entidades e Relacionamentos [Chen 76].

- ♦ **Mundo da documentação** - Reúne o conhecimento sobre o hiperdocumento em si. É formado por dois domínios:

Documentação das estruturas estáticas - Engloba o conhecimento sobre as estruturas que permanecem constantes enquanto o hiperdocumento está sendo usado.

Documentação das estruturas dinâmicas - Engloba o conhecimento sobre as estruturas que são alteradas durante o processo de navegação no hiperdocumento.

A utilização em conjunto destes domínios fornece ao usuário uma interpretação informal do conteúdo que foi formalmente representado no mundo do assunto.

- ♦ **Mundo da Apresentação** - Reúne o conhecimento sobre a maneira em que o sistema apresenta o hiperdocumento ao usuário, através dos artefatos de interface utilizados.

A partir da representação em mundos de assunto, documentação e apresentação foi desenvolvido um modelo generalizado que serve como base para qualquer hiperdocumento. Este modelo é representado pela figura 3.6. O processo de autoria se baseia no preenchimento deste modelo com informações sobre o domínio de uma aplicação específica. À medida em que este processo vai avançando, os autores do hiperdocumento vão instanciando e especializando as estruturas previstas por este modelo mais geral, até que se tenha uma representação particularizada para uma determinada aplicação.

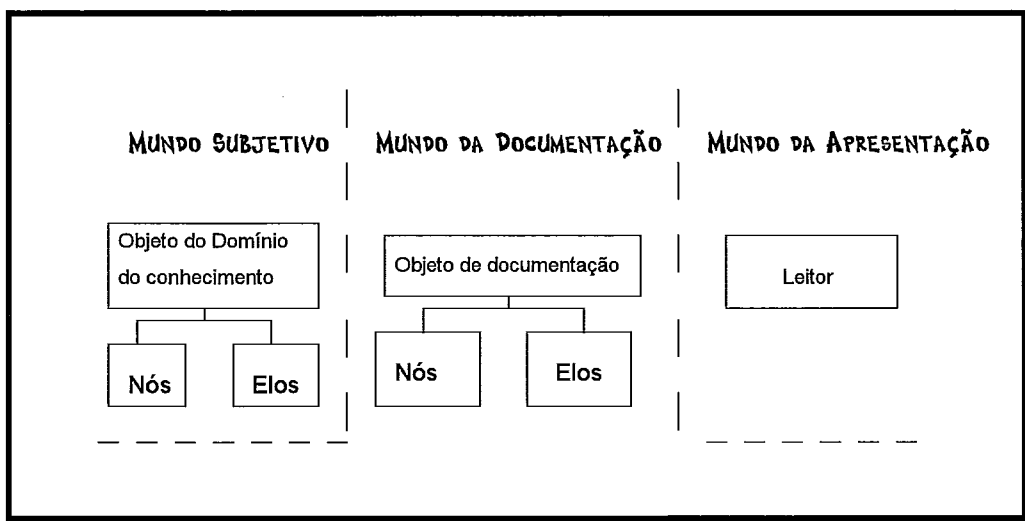


Figura 3.6 Conjunto de estruturas pré determinadas a serem preenchidas pelos autores de sistemas de hipertexto que estejam utilizando o ThyDoc

ThyDoc é um método de modelagem em ponto grande ("Authoring-in-the-large") de sistemas de hipertexto/ hipermídia que fornece equipe de autores uma base para a construção de um modelo generalizado do conteúdo da aplicação. O ponto de vista, baseado em conhecimento, permite que a informação seja melhor estruturada,

facilitando o entrosamento entre os participantes da equipe. A separação em conteúdo, estrutura e apresentação simplifica a complexidade da autoria em ponto grande.

3.2.2.3 O método HDM [Garz 90], [Garz 91]

O termo HDM - Hypermedia Design Model - é usado para designar um método para a criação do modelo de um sistema de hipertexto/hipermídia, baseado em uma linguagem abstrata e conceitual criada para descrever estruturas existentes em hiperdocumentos. O objetivo desta linguagem é a descrição de estruturas comuns a uma mesma família de hiperdocumentos. Para tal, o HDM fornece conceitos básicos de linguística para a especificação de propriedades semânticas e estruturais de uma aplicação em hipertexto ou família de aplicações, desconsiderando a implementação a ser utilizada.

Esta linguagem permite que os autores descrevam a estrutura da aplicação em um nível geral e abstrato, sem que haja necessidade de se especificar os nós individualmente.

Descreveremos, sucintamente, o funcionamento do HDM, para que logo a seguir possamos introduzir, em maior detalhe, cada um dos conceitos apresentados. O método HDM representa o domínio de uma aplicação através de *entidades* e *componentes*. As *entidades* podem assumir diferentes *tipos* e são compostas de hierarquias de *componentes*. Os *componentes* podem ser instanciados através de diferentes *perspectivas*. Tanto as *entidades* quanto os *componentes* podem se relacionar a outras *entidades* e *componentes* através de *elos*.

O conjunto dos elementos descritos acima forma um *esquema*. Um *esquema* de HDM é, portanto, um conjunto generalizado de definições, tais como *entidades* ou *elos*. A instância de um *esquema* em HDM é um conjunto particular destes elementos, definidos de acordo com uma aplicação específica. A seguir apresentaremos a descrição de cada um dos conceitos apresentados acima:

◆ **Entidades** - Uma entidade é um (grande) grupo de informações que representa um objeto concreto ou conceitual do mundo real, que pertence ao domínio da aplicação. De maneira mais geral, uma entidade é uma seção de um hiperdocumento que descreve um único tópico. Também podem ser definidos tipos de entidades, que são grupos de entidades que possuem propriedades em comum.

◆ **Componentes** - É bastante comum encontrarmos em aplicações do tipo hipermídia um conjunto de nós que mostram um mesmo tópico observado sob vários pontos de vista. Este conjunto de nós são chamados de componentes desta aplicação. Podemos defini-los mais formalmente como parte de uma informação que descreve algo sobre uma entidade.

◆ **Perspectivas** - São os diversos pontos de vista que podem ser aplicados a um mesmo tópico.

◆ **Unidades** - Nome dado a um conjunto de nós.

◆ **Elos** - Fazem a ligação entre informações contidas na base de dados da aplicação.

◆ **Contornos** - O HDM emprega este termo para designar porções de um hiperdocumento que contém informações para navegação do mesmo, mapas ou excursões. Como qualquer outra entidade modelada através do HDM, os contornos são organizados de maneira hierárquica.

◆ **Esquemas** - Pode ser definido como um conjunto de tipos de entidades e tipos de elos. Um esquema caracteriza classes de aplicações porque não chega a especificar as entidades e elos. Ao invés disto ele determina propriedades gerais para entidades em potencial e possíveis elos entre elas. Um esquema referente a uma aplicação específica é chamado esquema instanciado e se refere a instâncias das entidades e elos determinados no esquema geral.

Entidades e elos são elementos abstratos que o HDM utiliza para representar objetos conceituais. Durante a criação de um hiperdocumento, através da utilização de um *esquema*, estas *entidades* abstratas serão implementadas como nós. A figura 3.7 mostra um exemplo de modelagem do software chamado Nona Sinfonia de Beethoven, usando HDM. Este hipertexto serve como guia para a obra de Beethoven. É composto por mais de seiscentos nós, incluindo informações culturais e históricas sobre a vida e obra do compositor além de considerações acerca de seu trabalho e análise de sua obra.

As entidades Notas Históricas/Musicais descrevem os aspectos sociais e históricos da vida de Beethoven, interpretando a nona sinfonia e,

fornecendo comentários sobre o trabalho. Esta informação está disposta em duas perspectivas: texto e música.

Instâncias da Entidade Exemplo Musical podem ocorrer através das perspectivas música para CD, música para MAC (entenda-se, computadores do tipo McIntosh) e notação musical. Ainda são oferecidos dois tipos de contorno: contorno de assunto, que oferece uma sequência de tópicos que se referem a uma única referência histórica ou musical e o contorno musical que contém uma listagem de passagens relacionadas.

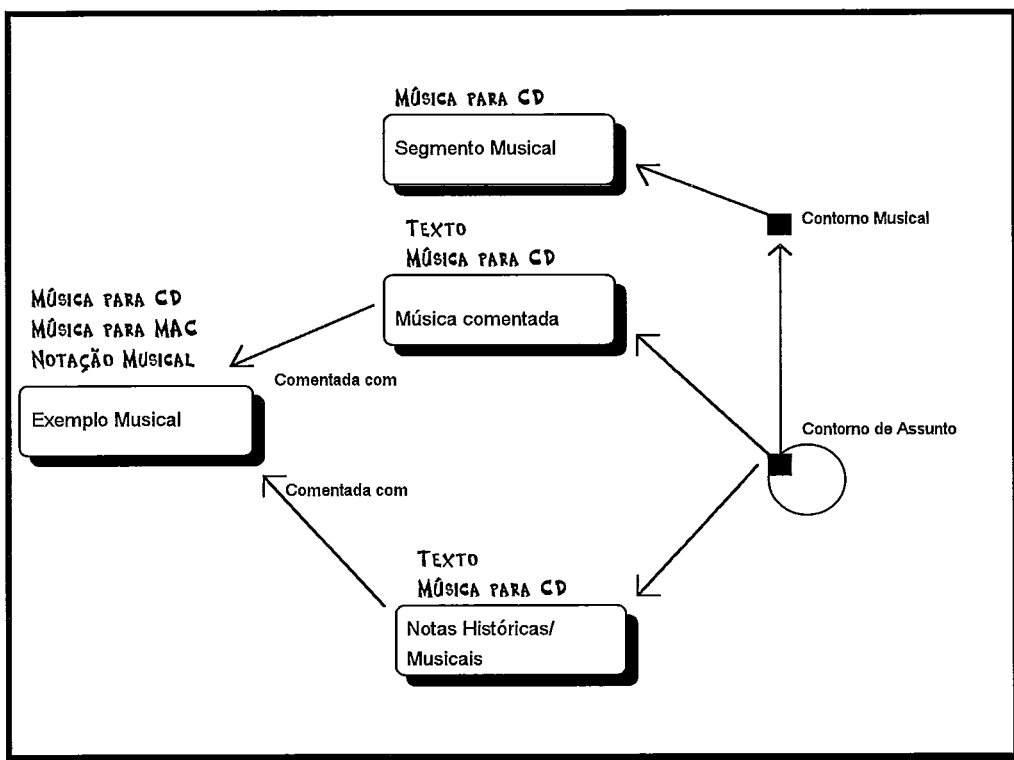


Figura 3.7 Esquema do Software Nona Sinfonia de Beethoven [Garz 91]

O HDM é um método de modelagem em ponto grande ("authoring-in-the-large") e, portanto, seu escopo está limitado estrutura da aplicação. O método não descreve a aparência da aplicação, isto é, os

tipos usados, a disposição das figuras, a maneira como são relacionados os elos, enfim nada relacionado com o conteúdo dos nós individuais. O método se propõe a descrever somente as relações estruturais e semânticas entre *grupos* de nós (o grifo é do próprio autor) [Garz 91].

3.3 Avaliação da Qualidade

Independente da classe de métodos utilizados para a autoria de aplicações hipertexto/hipermídia, um aspecto fundamental para se atingir o objetivo de gerar produtos que de fato atendam as expectativas que motivaram sua construção é realizar um controle da qualidade efetivo durante o processo de desenvolvimento da aplicação e ao final do mesmo.

Não se encontram na literatura trabalhos que abordem a avaliação de sistemas hipermídia ao longo de seu processo de desenvolvimento. Existem, entretanto, trabalhos que se referem qualidade das ferramentas de implementação e estudos no sentido de avaliação da qualidade de aplicações do tipo hipertexto/hipermídia. Descreveremos nas próximas seções dois trabalhos realizados na COPPE/ Sistemas.

3.3.1 Avaliação da qualidade de ferramentas para implementação de aplicações do tipo hipertexto

Mendonça [Mend 93] aborda o tema da avaliação de sistemas de hipertexto instanciando um método de avaliação da qualidade [Roch 87] e identificando um conjunto de fatores e critérios. Estes fatores

devem ser satisfeitos em um determinado grau, de maneira a alcançar o objetivo de *utilizabilidade* do sistema, garantindo, desta forma, a qualidade do produto final. Os objetivos determinados por cada fator são atingidos através de subfatores, que por sua vez se constituem de critérios, atributos primitivos que podem ser avaliados.

O trabalho apresenta, portanto, um conjunto de fatores adequados avaliação de ferramentas para implementação de aplicações do tipo hipertexto. A análise da qualidade das ferramentas disponíveis, apoia o processo de seleção do software a ser usado na implementação de uma aplicação do tipo hipertexto/hipermídia, garantindo a qualidade do produto final. Apresentaremos sucintamente os fatores identificados: utilizabilidade para a autoria, utilizabilidade para a leitura, manutenibilidade, operacionalidade, portatibilidade, rentabilidade e cooperatividade, que definiremos a seguir.

◆ **Utilizabilidade para a autoria** - Caracteriza a capacidade da ferramenta de prover facilidades para a realização da atividade de autoria de um hiperdocumento.

◆ **Utilizabilidade para a leitura** - Caracteriza a capacidade do programa de prover facilidades para a realização da atividade de leitura de um hiperdocumento.

◆ **Manutenibilidade** - Caracteriza a capacidade da ferramenta de permitir a realização de alterações em hiperdocumentos já implementados.

◆ **Operacionalidade** - Caracteriza a capacidade da ferramenta de ser amena ao uso, facilitando a comunicação com o usuário.

- ◆ **Portatibilidade** - Caracteriza a capacidade da ferramenta de poder ser operada com facilidade e adequadamente em plataformas diferentes daquela originalmente planejada.
- ◆ **Rentabilidade** - Representa a característica do sistema se possuir relação custo/benefício aceitável.
- ◆ **Cooperatividade** - Caracteriza a capacidade da ferramenta de apoiar a realização de trabalho cooperativo.

A figura 3.8 mostra os fatores definidos acima, apresentando os critérios relacionados a cada um.

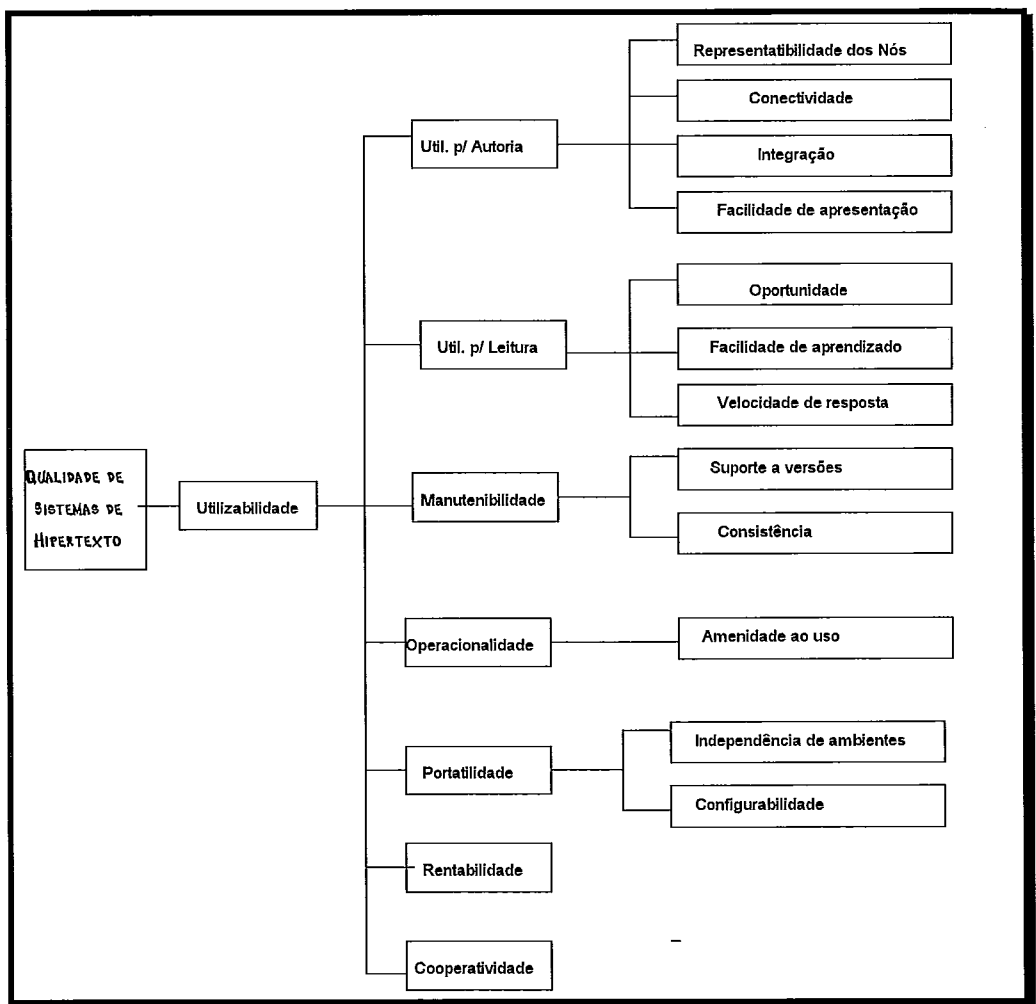


Figura 3.8 Qualidade de sistemas de Hipertexto [Mend 93]

3.3.2 Avaliação da Qualidade de aplicações do tipo Hipertexto/Hipermídia

O tema da avaliação da qualidade de aplicações do tipo hipertexto/hipermídia é abordado por Mendonça [Mend 93b] através da ótica da *operacionalidade*. Esta é definida como a característica de uma aplicação de hipertexto ser amena ao uso, facilitando a comunicação entre o usuário durante todo o tempo em que este o utilizar [Roch 87].

A amenidade ao uso é a característica de uma aplicação em hipertexto apresentar os resultados esperados, oferecendo sentimento de satisfação e de confiança ao usuário. É avaliada através dos seguintes critérios: exibição de mensagens e comandos em português, visualização dos botões, suporte ao rolamento do conteúdo de uma janela ou tela, adequação do estilo de interface, adequação do grau de integração entre autoria e navegação, ocupação da tela, personalização e restauração.

Os critérios mencionados estão mostrados na figura 3.9

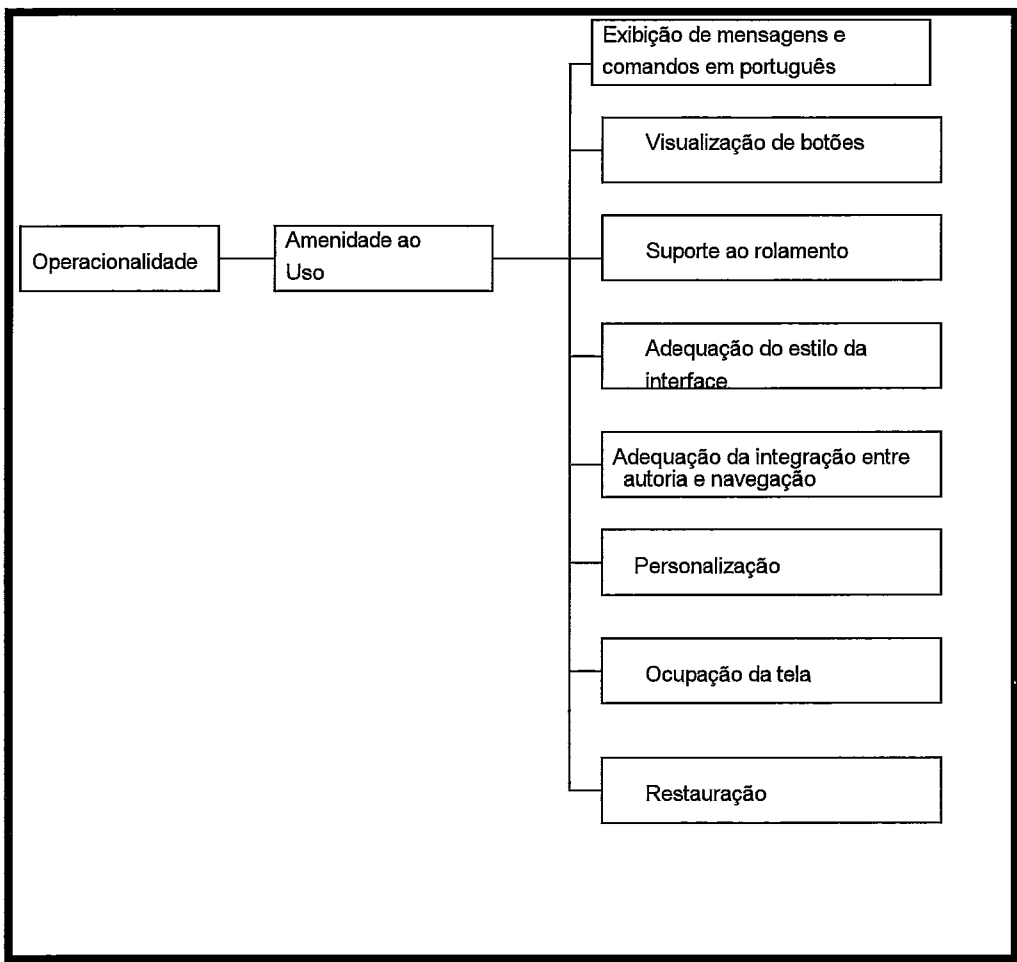


Figura 3.9 Critérios utilizados para a avaliação da qualidade de aplicação do tipo hipertexto/ hipermídia

A principal vantagem da utilização de um método para a avaliação da qualidade é fornecer critérios quantificáveis que permitam que toda a equipe envolvida no desenvolvimento da aplicação avalie o produto segundo uma mesma ótica.

3.4 Conclusão

Os métodos abordados neste capítulo tratam, de maneira geral da autoria em ponto grande. Este tipo de abordagem tem como objetivo principal fornecer as linhas gerais da estrutura de aplicações do tipo hipertexto/hipermídia compostos por um grande número de nós, onde é possível identificar-se estruturas repetidas que podem ser padronizadas, como é o caso das entidades abordadas pela metodologia HDM.

Estes métodos, entretanto, não resolvem totalmente o problema da autoria em hipermídia. São necessários métodos que apoiem a especificação detalhada de cada nó, descrevendo seu conteúdo e sua forma, isto é a maneira com que este será apresentado ao usuário final (por exemplo, cor, fonte, disposição das figuras, entre outras). Necessita-se, portanto, de métodos que apoiem a autoria em ponto pequeno. Estes métodos, também, devem ser capazes de mapear as ligações existentes entre os nós, permitindo que o sistema seja implementado em qualquer ferramenta e, facilitando futuras manutenções no sistema.

Outro aspecto fundamental para a implementação de sistemas hipermídia é dispor-se de procedimentos e critérios para a avaliação da qualidade, não apenas do produto final mas, sobretudo, ao longo do processo de desenvolvimento.

No próximo capítulo apresentaremos a proposta de um método para a especificação e projeto de aplicações hipermídia, Hiper Autor, que apoia tanto a autoria em ponto pequeno quanto em ponto grande.

O método Hiper Autor faz parte de um ambiente mais amplo, em desenvolvimento na COPPE, dedicado ao desenvolvimento de aplicações hipermídia envolvendo não apenas os aspectos de construção do produto tratados pelo Hiper Autor mas, também, aspectos gerenciais e de controle da qualidade.

CAPÍTULO 4

O Método Hiper Autor



4.1 Justificativas

Tem se observado duas tendências ao se considerar as características do pessoal envolvido na elaboração de software educacional. Numa primeira opção, o desenvolvimento é realizado por profissionais das áreas de conhecimento para as quais se deseja produzir as aplicações. Estes profissionais, de maneira geral, possuem pouca ou nenhuma experiência em lidar com sistemas de computação. A segunda opção é realizar o desenvolvimento com a equipe constituída somente por profissionais da área de Informática (analistas, programadores e engenheiros de software) que não possuem conhecimentos específicos da área de aplicação mas contam com as habilidades necessárias para a elaboração de um produto de software.

Como apontamos no Capítulo 1, tanto uma quanto outra opção não produzem, via de regra, resultados satisfatórios.

4.2 Objetivos do HiperAutor

Uma solução natural para esta dificuldade é a adoção de equipes multidisciplinares. Entretanto, uma das maiores restrições relacionadas ao desenvolvimento de software é o modo segundo o qual é formada a equipe de trabalho. Esta problemática é inerente a tentativa de se coordenar profissionais de áreas diferentes em torno de um mesmo objetivo.

Visando minimizar estes problemas e aumentar a cooperação entre os membros da equipe, decidimos adotar o enfoque de projeto participativo proposto por Morten Kyng, que considera a interação entre os usuários e desenvolvedores ao longo do processo de desenvolvimento de software como um aprendizado mútuo, ou seja, uma estratégia que implica no aprendizado do domínio da aplicação por parte dos desenvolvedores e no aprendizado sobre os aspectos tecnológicos por parte dos usuários. [Kyng 91]. Seu objetivo é criar um ambiente onde o produto possa ser produzido de maneira eficaz e criativa, através da troca de conhecimentos e experiências anteriores de pessoas com diferentes perfis. Holden afirma que a construção de sistema do tipo hipermídia, a partir da autoria compartilhada entre diferentes autores, alunos e professores, tem sido apontada como um dos caminhos mais promissores para a cooperação entre alunos em sala de aula [Hold 92]. Segundo Campos, devemos ressaltar no trabalho cooperativo aplicado ao desenvolvimento de software educacional, a importância de:

- ◆ dar a cada participante a liberdade de registrar e associar idéias durante a autoria;
-

- ◆ decidir em grupo em que ordem abordar os temas e como escolher suas partes;
- ◆ atuar em múltiplas frentes de modo a facilitar o registro imediato de idéias. [Camp 92].

Entendemos [Roch 92] que uma equipe multidisciplinar, com enfoque de projeto participativo, reunindo profissionais de várias áreas é a solução adequada para o problema da autoria e implementação e a única possibilidade de se ter um produto de software de boa qualidade. Desta maneira propomos que o desenvolvimento de aplicações do tipo hipertexto/hipermídia, com qualquer finalidade, deve ser feito por uma equipe multidisciplinar, que conterà profissionais da área de informática, especialistas da área de interesse da aplicação e pedagogos. Segundo Perez, [Pere 91], esta equipe deverá ainda conter, dependendo da natureza e usuários do produto, outros especialistas como por exemplo psicólogos, bibliotecários e comunicadores visuais.

Nesta equipe multidisciplinar, cada membro tem sua função de acordo com suas habilidades e experiências. A participação de cada um, ao longo do desenvolvimento, se dá nos momentos onde as atividades em questão requeiram suas habilidades específicas.

HiperAutor é um método para a especificação que faz parte de um ambiente para desenvolvimento de software especialmente definido para apoiar o trabalho deste tipo de equipe ao desenvolver aplicações educacionais do tipo hipertexto/hipermídia. A estratégia de desenvolvimento, prevista no método, auxilia os especialistas em

computação a realizar a elicitação de requisitos do sistema junto aos especialistas do domínio da aplicação.

Ao longo deste processo, os dados elicitados são modelados, de maneira a gerar uma representação gráfica passível de ser avaliada em qualquer momento pela equipe.

A seguir serão apresentados o ciclo de vida e o método de desenvolvimento do Hiper Autor.

4.3 O Ciclo de Vida

O termo ciclo de vida é, normalmente, empregado para definir todas as etapas do processo de desenvolvimento de software, especificando todas as atividades que devem ser realizadas em cada uma destas etapas [Roch 87]. Seu conceito embute a idéia de que, desde sua concepção inicial até sua implementação, um sistema sofre um desenvolvimento gradual [Ghe 91]. O ciclo de vida de um software engloba, de maneira geral, todas as atividades necessárias para sua definição, desenvolvimento, implementação e operação.

Métodos visam auxiliar a execução de uma ou mais atividades previstas no processo de desenvolvimento de um software. As características específicas de cada método vão determinar sua adequação a um determinado modelo de ciclo de vida. Outros aspectos como características do projeto e o grau de conhecimento da equipe de desenvolvedores sobre o domínio da aplicação, o hardware

e o software a serem utilizados, também, podem determinar o modelo de ciclo de vida mais adequado ao desenvolvimento numa determinada circunstância [Agui 92].

Existem, na literatura, diversas propostas para modelos de ciclo de vida. Estas propostas concatenam diversos aspectos que, sob cada uma das abordagens, representam o conjunto de atividades e controles necessários para a elaboração de um software.

Na próxima seção descrevemos alguns destes modelos de ciclos de vida para evidenciar o mais adequado para nossa proposta de ambiente da qual faz parte o Hiper Autor.

4.3.1 Modelo de Ciclo de Vida Clássico

É o modelo tradicional de desenvolvimento de software, também conhecido como modelo em cascata. Divide o processo de elaboração do software em uma série de fases sucessivas. Cada uma destas fases possui um ponto de partida e de chegada bem definidos e deve produzir um determinado resultado para que a próxima fase possa ser iniciada [Ghez 91]. Na realidade, cada uma das fases é um refinamento da fase anterior [Fair 85].

O ciclo de vida clássico demanda uma abordagem sistemática no processo de desenvolvimento do software, que começa no nível macroscópico de sistema e compreende as fases de análise, projeto, construção, codificação, avaliação e manutenção [Pres 92].

Outros autores dividem de maneira diferente as fases que compõe o ciclo de vida tradicional. Yourdon propõe sete fases: requisitos do sistema, requisitos do software, análise, projeto de programas, codificação, testes e operação [Your 89]; Bersoff e Davis dividem a fase de projeto em duas: projeto preliminar e projeto detalhado [Berk 91]. Estas diferenças, entretanto, não são significativas e representam pequenas variações do mesmo modelo. A figura 4.1 mostra o ciclo de vida clássico.

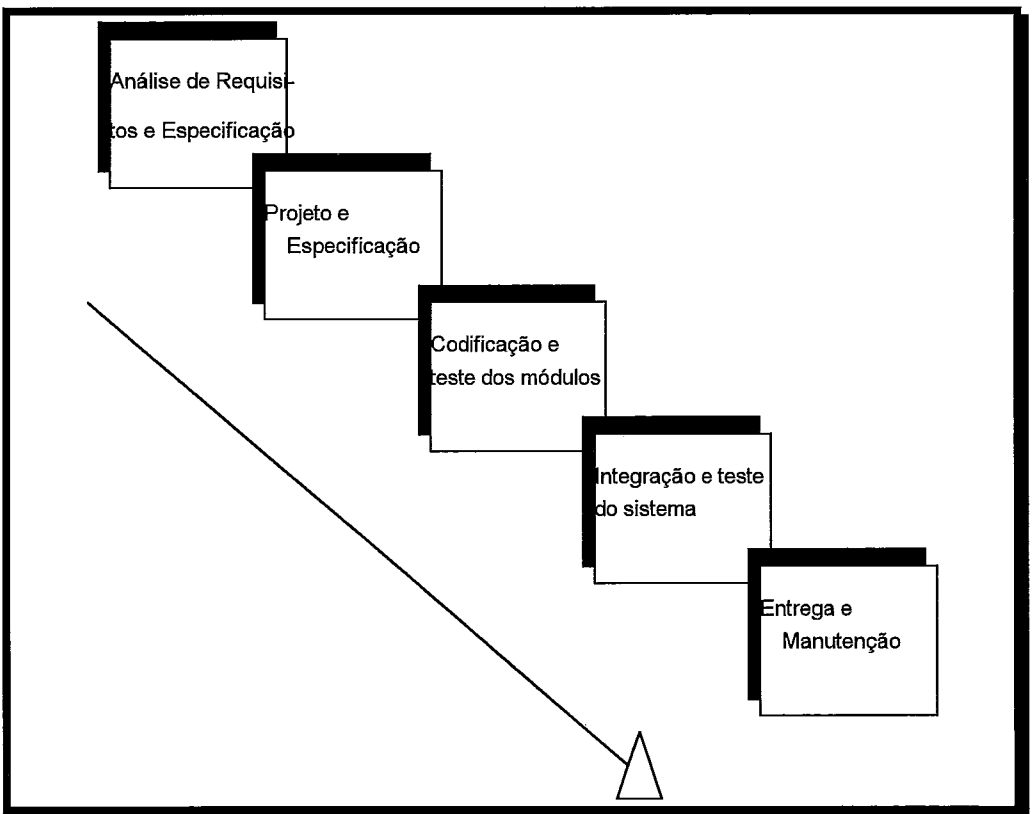


Figura 4.1 O ciclo de vida Clássico [Ghe 91]

Na realidade o que caracteriza o ciclo de vida clássico é "a progressão linear e sequencial entre uma fase e a seguinte" [Your 89],

levando a passagem para próxima fase somente quando a anterior estiver totalmente completa. Esta característica faz com que a qualidade dos produtos de uma fase seja dependente da fase anterior, ou seja, a qualidade da fase de projeto depende da qualidade da fase de análise e assim por diante. Outra característica desta abordagem é atrasar a detecção de erros para a fase final, pois no modelo clássico a verificação do sistema, sob forma operacional, só é realizada após sua implementação completa [Your 92].

Estas características do modelo implicam que uma versão operacional do produto só seja mostrada ao usuário nas etapas finais do ciclo de vida , quando a fase de codificação estiver terminada [Pres 92] dificultando a avaliação do produto ao longo do desenvolvimento o que pode ter fortes incidências sobre os custos e o próprio sucesso de um projeto.

4.3.2 O Modelo de ciclo de vida baseado em protótipos

"Prototipagem é a técnica de construção de uma implementação parcial de um sistema de modo a permitir que usuários e desenvolvedores possam aprender mais sobre um determinado problema ou sobre a solução para o mesmo. " [Dav 90]

Fala-se em implementação parcial pois, se considerarmos uma implementação completa, esta seria o próprio sistema, não seu protótipo. Deve-se notar que existe uma diferença entre implementação parcial e implementação de um subconjunto do

sistema: o objetivo da primeira é o aprendizado relativo ao domínio do problema enquanto que o objetivo da implementação de subconjuntos de um sistema é prover funcionalidade.

Pressman [Pre 92] afirma que a prototipagem capacita o desenvolvedor a modelar o software a ser construído. Este modelo pode tomar três formas: (1) um protótipo em papel que dispensa a implementação da interface homem-máquina, desde de que esta fique clara para o usuário; (2) um protótipo operacional que implemente um subconjunto do sistema; (3) um programa já existente, que seja capaz de realizar parte das funções desejadas do sistema. A figura 4.2 mostra o ciclo de vida baseado em prototipagem.

Segundo Davis existem duas abordagens para prototipagem: evolutiva e descartável [Davi 90]. Na prototipagem descartável é construída uma implementação parcial do produto, antes ou durante a fase de especificação do produto, contendo os aspectos menos entendidos do sistema. Protótipos descartáveis são construídos de maneira rápida, e são usados para validar requisitos, obter a experiência necessária para a determinação de novos requisitos ou validação do projeto e são descartados no final do processo [Ber 91] . Depois que o protótipo é completado, o desenvolvedor escreve a Especificação de Requisitos do software, incorporando o que foi aprendido e constrói o sistema baseado nesta especificação.

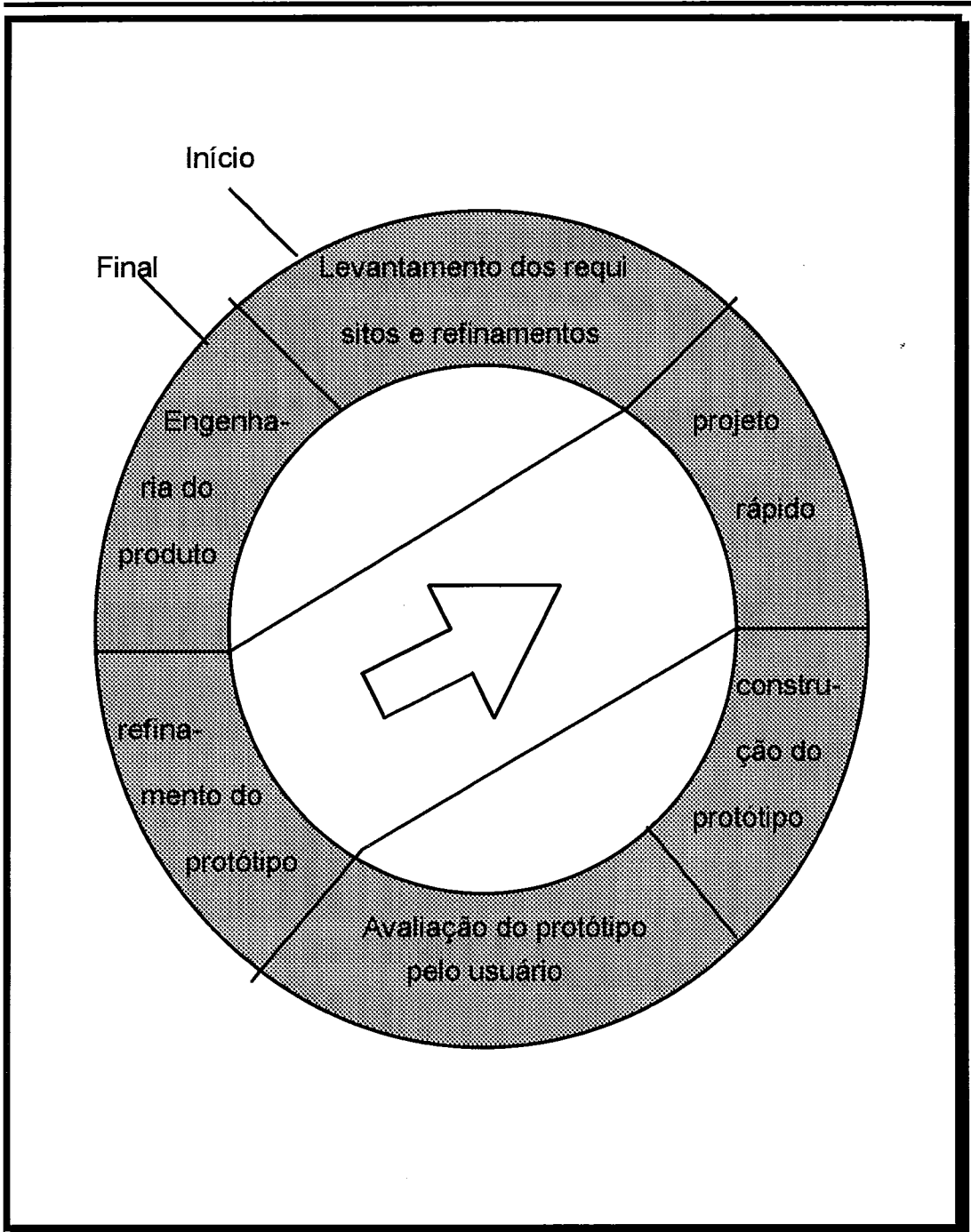


Figura 4.2 Ciclo de Vida baseado em prototipagem [Pre 92]

Protótipos descartáveis são úteis e adequados na verificação de pequenas partes de sistemas complexos [Davi 92]. A Figura 4.3 mostra o modelo de ciclo da vida de software utilizando a prototipagem descartável.

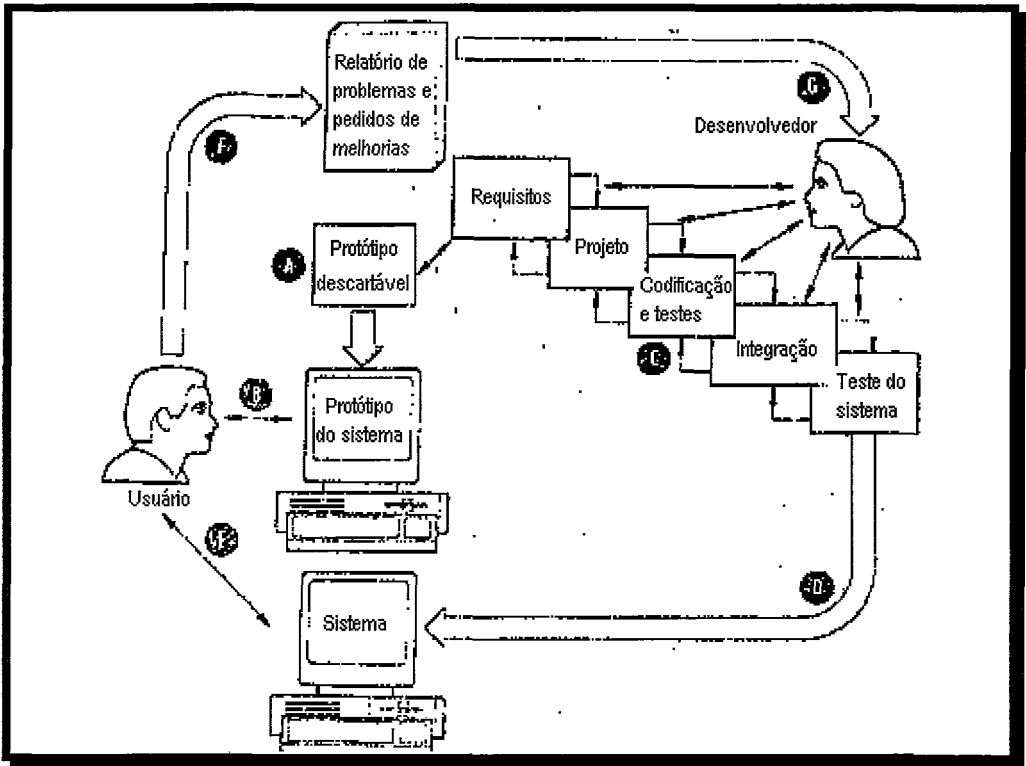


Figura 4.3 O modelo de ciclo de vida de software utilizando a prototipagem descartável [Davi 92]

Protótipos evolutivos são programas de alta qualidade usados para validar supostos requisitos, ganhar a experiência necessária para identificar novos requisitos ou validar um possível projeto e são repetidamente modificados quando um novo dado é aprendido [Ber 91]. Quando uma nova versão é completada, o desenvolvedor modifica a Especificação de Requisitos do software de modo a incorporar as mudanças. Protótipos evolutivos funcionam bem quando a maioria das funções críticas são conhecidas. Na realidade, de acordo com esta abordagem desenvolve-se em cada momento o "todo", de início de forma bem simplificada mas que vai ganhando detalhes a cada nova versão. A figura 4.4 mostra o modelo de ciclo de vida de software utilizando a prototipagem evolutiva.

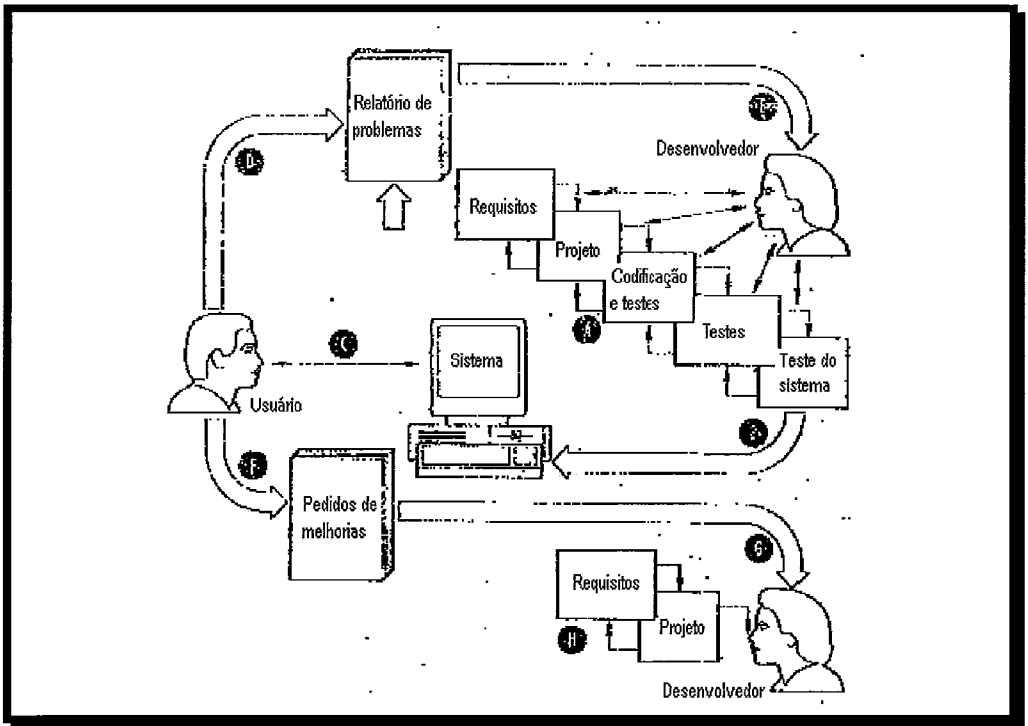


Figura 4.4 O modelo de ciclo de vida de software utilizando a prototipagem evolutiva [Davi 92]

Proposta mais recente mostra a tendência para uma nova modalidade de prototipagem: a prototipagem operacional [Davi 92]. Esta abordagem se propõe resolver situações onde nem a prototipagem evolutiva nem a descartável sozinhas são as mais adequadas. Estas situações acontecem, tipicamente, quando os requisitos do sistema a ser construído são críticos ao projeto e bem entendidos, ou não são críticos ao projeto mas são mal compreendidos. Neste caso, a prototipagem operacional combina as principais e melhores características tanto da prototipagem evolutiva quanto da descartável [Ber 90], funcionando da seguinte forma: (1) constrói-se uma base sólida para o desenvolvimento do software, baseado em conceitos de engenharia de software pois parte-se de uma abordagem de prototipagem evolutiva; (2) o protótipo operacional é levado aos futuros usuários juntamente com um prototipador que fará,

rapidamente, as mudanças requisitadas pelos primeiros e construirá um protótipo descartável atendendo a estes pedidos; (3) depois que o protótipo é avaliado e aprovado junto ao usuário, incorporam-se as novas mudanças na versão do software anteriormente construída, abandonando-se o protótipo descartável. A figura 4.5 mostra o modelo de ciclo de vida de software utilizando a prototipagem operacional.

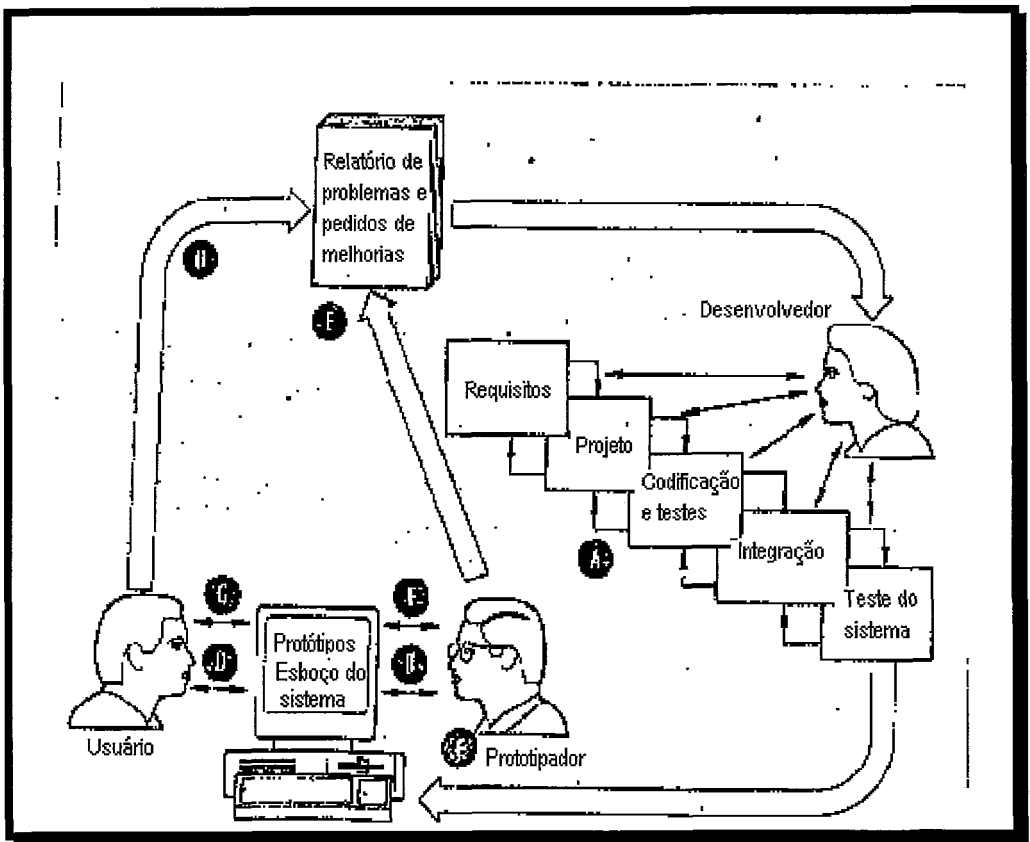


Figura 4.5 o modelo de ciclo de vida de software utilizando prototipagem operacional [Davi 92].

Independente da forma escolhida, a prototipagem visa facilitar a comunicação entre usuários e desenvolvedores. Boar [Boar 84] aponta as vantagens da utilização da prototipagem na fase de Análise de Requisitos como se segue: " Uma abordagem alternativa para a definição de requisitos é obter um conjunto inicial de necessidades e implementá-las rapidamente com a intenção declarada de expandí-las e refiná-las interativamente à proporção do aumento de conhecimento mútuo do sistema por parte do usuário e desenvolvedor. " Davis [Dav 90] quando enumera os impactos causados pelo uso de prototipagem na fase de Análise de Requisitos, enumera as seguintes vantagens da apresentação de um protótipo a um futuro usuário: (1) determinação da possibilidade de implementação dos requisitos; (2) validação da real necessidade de determinadas funções; (3) determinação da falta de alguns requisitos, e, (4) determinação da viabilidade das interfaces com os usuários. Finalmente, ao comentar técnicas de levantamento de requisitos, Yourdon [You 92] coloca que, " a *única* maneira de se elicitar conhecimento de um usuário que não tenha certeza sobre a natureza do sistema que deseja construir, é através da prototipagem". (O grifo é do próprio autor)

4.3.3 O Modelo de Ciclo de Vida em Espiral

Também conhecido como ciclo de vida em cascata moderno, este modelo proposto por Boehm [Boeh 88] engloba o melhor tanto do modelo de ciclo de vida clássico quanto do baseado em prototipagem [Pres 92]. O modelo sugere que o desenvolvimento de um produto de software pode ser realizado através de uma série de incrementos,

podendo, desta maneira, se construir uma série de modelos de ciclo de vida clássico, um para cada incremento [Your 92]. Esta série de incrementos está organizada em forma de espiral como mostra a figura 4.6

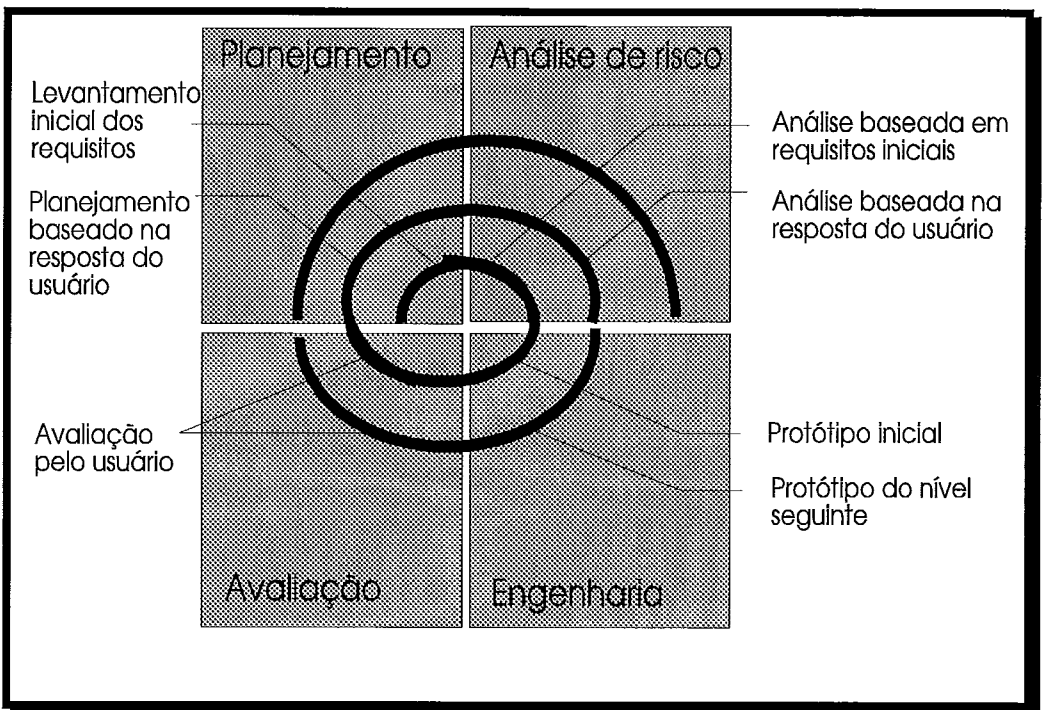


Figura 4.6 - O Modelo Espiral [Pres 92]

Cada interação consiste em uma volta completa em torno da espiral, começando no centro e se dirigindo para fora, sendo que em cada uma delas uma nova versão do software é construída. O desenvolvimento de cada uma das interações é baseado no cumprimento das seguintes atividades [Pres 92]:

1- Planejamento: que implica em determinação de objetivos, alternativas e limitações;

- 2- Análise de Risco: que implica em análise das alternativas e identificação/resolução dos riscos;
- 3- Engenharia: que implica em elaboração da próxima versão.
- 4- Avaliação: pelo usuário- que implica na aprovação da última versão;

Durante o primeiro circuito em volta da espiral são definidos os objetivos e alternativas. Limitações e riscos são identificados e analisados. Se a fase de análise de riscos determinar incertezas, quanto aos requisitos do projeto, são usadas técnicas de prototipagem junto ao usuário e analistas para redefiní-las. Na fase de engenharia é construída uma versão do software especificado. Esta versão será então validada pelos usuários na próxima fase, onde estes farão sugestões para modificações. Estas informações permitirão uma nova análise de requisitos para o projeto, iniciando a segunda interação no modelo. Desta forma, serão feitas versões sucessivas do software na direção da obtenção de um modelo cada vez mais completo, até que se alcance a forma definitiva, validada pelos usuários do sistema.

O modelo de ciclo de vida em espiral é, atualmente, a aproximação mais real para o desenvolvimento de sistemas de software [Pres 92]. Engloba o uso da prototipagem para redução de riscos e ao mesmo tempo habilita o desenvolvedor a se utilizar da mesma para aumentar a interação com o usuário. Esta interação é realizada através das sucessivas versões do software que facilitam a avaliação por parte do usuário, dos requisitos que ou estão faltando à especificação ou não estão definidos corretamente.

Considerando os modelos contemplados anteriormente e, tendo em vista tirar proveito das vantagens oferecidas pelo modelo baseado em prototipagem utilizaremos como modelo de ciclo de vida para o HiperAutor o modelo em espiral.

4.4 Método de desenvolvimento

O método de desenvolvimento HiperAutor estabelece uma forma de organizar e disciplinar o pensamento e o trabalho da equipe desenvolvedora, estabelecendo para cada etapa do processo de desenvolvimento técnicas e instrumentos adequados para sua realização. Para alcançar estes objetivos, o método Hiper Autor conta com duas linguagens gráfico-linguísticas e formulários para documentação. Estas linguagens e formulários são utilizados pelos especialistas em computação durante o processo de sucessivas interações com os especialistas no domínio do problema. As linguagens e formulários apoiam na organização destas interações e na documentação do produto.

O desenvolvimento se dá segundo o modelo de ciclo de vida em espiral através da realização de uma série de etapas que estão relacionadas à construção do produto nas fases de especificação de requisitos e projeto. As fases previstas pelo método são: levantamento inicial dos requisitos, elaboração do esquema geral do conteúdo, elaboração da rede de nós, avaliação, especificação detalhada de

projeto, implementação da versão 1 e, avaliação, conforme ilustrado na figura 4.7.

Na figura 4.8 situamos estas atividades no ciclo de vida em espiral, que é o modelo adotado pelo Hiper Autor.

Hiper Autor											
Etapa						Responsável					
Levantamento inicial de Requisitos						especialistas do assunto analistas					
Elaboração do Esquema Geral do Conteúdo						especialistas do assunto analista					
Elaboração da Rede de Nós						especialistas do assunto analistas					
Avaliação						especialistas do assunto analistas					
Especificação detalhada do Projeto						especialistas do assunto analistas programador visual					
Implementação da versão 1						implementador					
Avaliação						especialista do assunto analista					
Novas Versões						implementador analistas					

Figura 4.7 Etapas previstas no HiperAutor

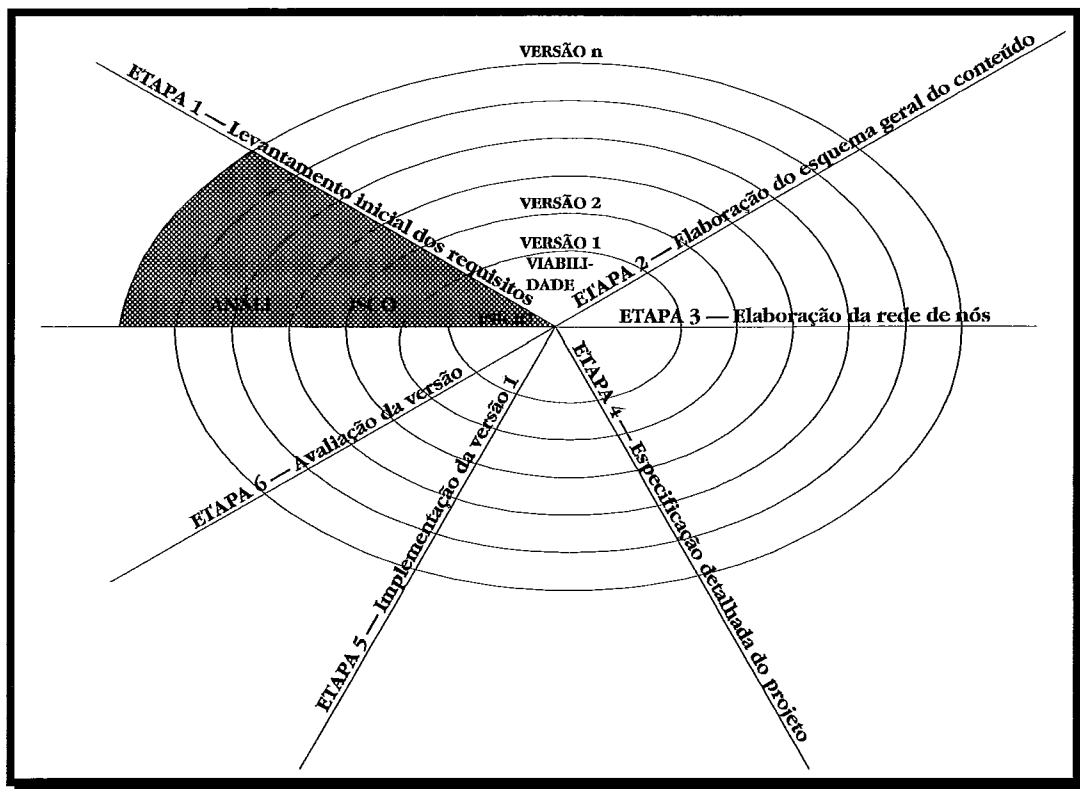


Figura 4.8 Ciclo de Vida do Hiper Autor

A seguir apresentaremos o detalhamento de cada uma das atividades previstas.

4.4.1 Levantamento Inicial dos Requisitos

A finalidade desta etapa é estabelecer o escopo da base de dados da aplicação e identificar os objetivos educacionais. Esta atividade deve ser realizada pelos especialistas do assunto e pelo menos um analista.

Neste momento é definido o escopo da aplicação, isto é, qual será o assunto e, em que profundidade será tratado pela aplicação.

Este processo inicia-se com a elicitação de requisitos junto aos especialistas. Entendemos por elicitar o processo de coletar, validar e tornar explícito os requisitos do software a ser construído. Se consideramos um requisito como condição necessária para obtenção de certo objetivo, podemos definir a primeira parte do processo de Especificação de Requisitos, a Análise do Problema como:

"Um processo em que o que deve ser feito é elicitado e modelado. Este processo deve lidar com diferentes pontos de vista, e usa uma combinação de métodos, ferramentas e atores. O produto deste processo é um modelo, do qual um documento, chamado requisitos, pode ser extraído. " [Leit 90]

O processo de elicitação compreende os processos de: coleta de fatos, validação dos fatos e comunicação. O processo de modelagem compreende os processos de organização e representação, como mostra a figura 4.7

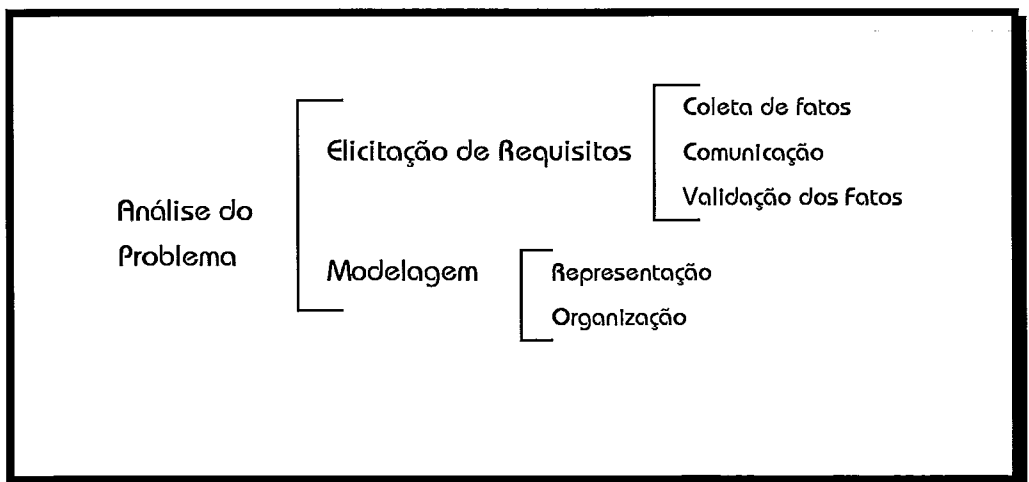


Figura 4.9- O processo de Análise do Problema [Leit 90]

É durante a elicitaco que o conhecimento do universo de informao de uma determinada aplicao ´ adquirido, para que depois possa ser modelado. Modelagem ´ o processo responsvel pela caracterizao dos resultados obtidos pela elicitaco de requisitos. Estes resultados formam o contedo da aplicao, chamado de base de dados.

Na atividade de modelagem so definidos os contextos. Chamamos de contextos a partio da base de dados em grandes grupos de assuntos afins. Consideramos adequado tratar at´ cinco contextos dentro de uma ´nica base de dados; um nmero superior no ser facilmente gerencivel. Provavelmente seria possvel unir contextos, nesta situao, a um dos outros contextos existentes. Em muitos casos tratamos de assuntos que aceitam uma subdiviso lgica de seus contextos; estas subdivises chamaremos de subcontextos. Para realizar estas atividades precisaremos dos especialistas no assunto e educadores, que, determinaro, juntamente com a equipe de analistas, a melhor diviso do tema desejado, de maneira a manter sua consistncia e alcanar os objetivos educacionais esperados. O resultado desta fase ´ um Diagrama de Contexto que ser representado em um formulrio como o da figura 4.10.

F1 Diagrama de Contexto

HA HiperAutor

Objetivos:

Figura 4.10 Formulário para Diagrama de Contexto

Para construção do diagrama de contexto tem-se uma linguagem gráfica composta de caixas (quadrados e retângulos) e arcos. A figura 4.11 mostra a notação gráfica.

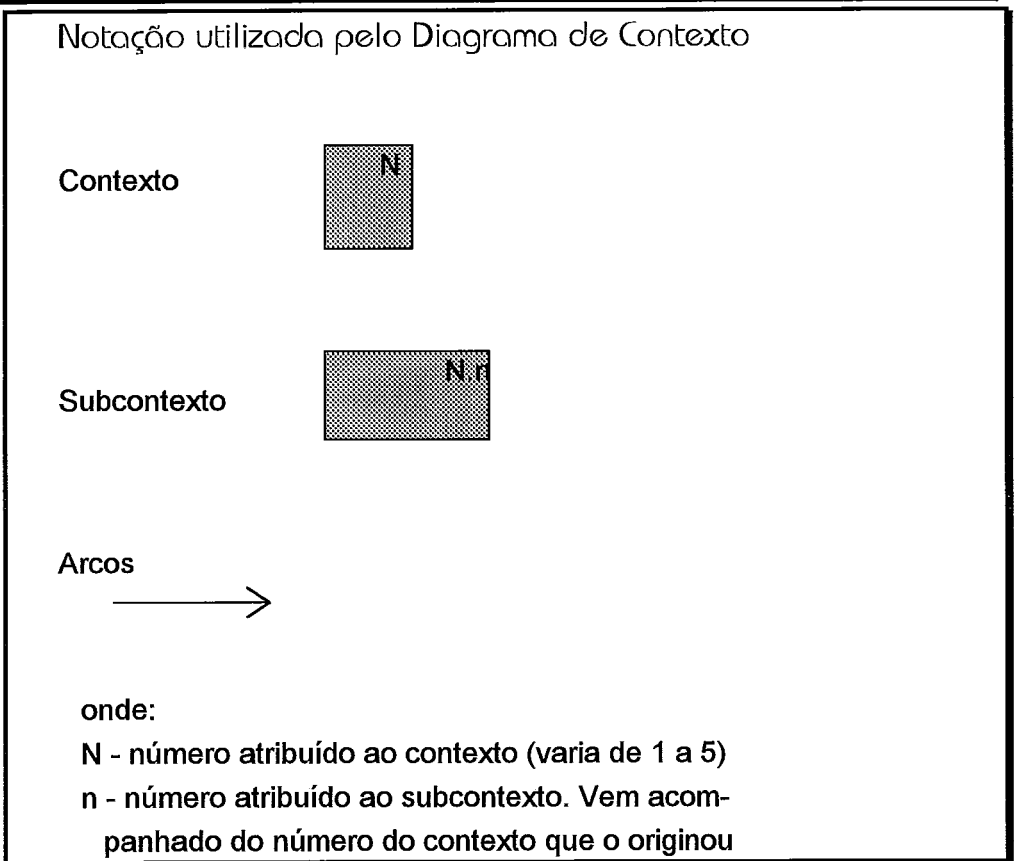


Figura 4.11 Notação utilizada pelo Diagrama de Contexto

A figura 4.12 mostra o diagrama em blocos para esta etapa.

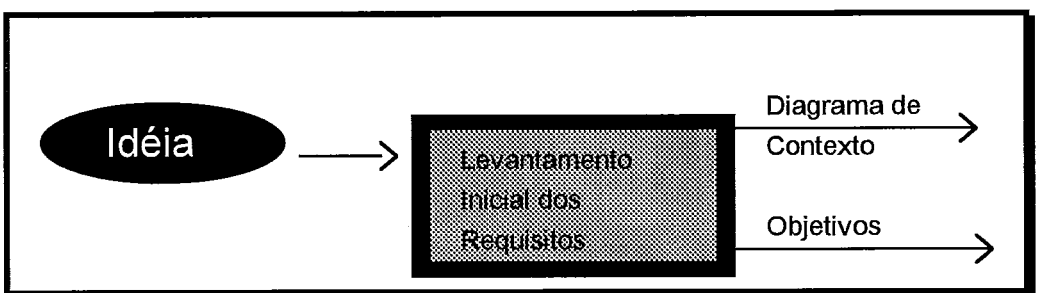


Figura 4.12 Diagrama em blocos para a etapa "Levantamento inicial dos Requisitos"

Para maior exemplificação da utilização deste formulário vide anexo.

4.4.2 Elaboração do Esquema Geral do Conteúdo

O objetivo desta etapa é fazer um esboço do que será o conteúdo da aplicação. O analista de sistemas, junto com os especialistas, faz um apanhado geral dos tópicos que vão compor cada um dos contextos definidos na fase anterior.

Nesta etapa não deve haver preocupação quanto à redação do texto ou precisão quanto ao detalhamento de imagens e sons a serem utilizados. Os autores - especialistas do assunto, devem se ater a fornecer somente uma descrição geral dos tópicos, figuras e sons, que desejam que faça parte da aplicação. A possibilidade e a forma de execução de tais pedidos será determinada mais adiante.


Glushko [Glu 89] afirma que os autores de sistemas de hipertexto devem ter em mente os seguintes objetivos: promover a acessabilidade da informação, aumentar a usabilidade e aumentar a satisfação do leitor.

Para favorecer o alcance destes objetivos, o HiperAutor prevê que se explicita, já desde este momento, o modo como a informação será interligada. A interconecção entre os tópicos é feita através de ligações, isto é, "um ponteiro que liga um tópico a outro" [Gay 91]. Acessamos ligações através de botões, "representação de uma ligação na tela do monitor" [Ber 91].

Nesta fase não é necessário elaborar, completamente, o conteúdo de cada contexto, mas já se deve fazer um levantamento inicial de como

os blocos de informação estão interrelacionados. Estes dados serão elicitados junto aos especialistas utilizando-se, para registro, o formulário 2 - Formulário Simplificado - (figura 4.13).

F2 Formulário Simplificado



HiperAutor

A

Texto

Conteúdo

Ligações (NomeBotão - NomeNóDestino)

A -	E -
B -	F -
C -	G -
D -	H -

Obs:/

Som

Descrição

Imagem/Vídeo

Descrição

Figura 4.13 Formulário Simplificado

As normas para o preenchimento do formulário são as seguintes:

- ◆ **Nome:** Atribuir um nome único e compatível com o tópico que está sendo tratado.
- ◆ **Número:** Repetir o número relativo ao contexto e subcontexto do qual este se derivou e atribuir uma extensão do tipo .n, onde n são números inteiros a partir de 1. Desta forma garantimos que para cada um dos nós haverá um código único que o distingue dos demais.
- ◆ **Texto:** Breve descrição dos tópicos a serem abordados sem preocupação com a redação.
- ◆ **Ligações:** fazer uma previsão das ligações que se deseja fazer a outros nós. Atribuir um número à ligação (número do nó corrente adicionado de uma extensão) e colocar o nome do contexto/subcontexto que se deseja referir.
- ◆ **Som:** Breve descrição do som (voz humana ou música) que se deseja inserir na aplicação. Não é necessário especificar a fonte do som ou detalhes sobre a digitalização do mesmo.
- ◆ **Imagem:** Breve descrição da(s) imagem(ns) que se deseja que faça(m) parte da aplicação. Não é necessário que esta imagem já exista, ela pode ser criada ou digitalizada se necessário for.
- ◆ **Observação:** Campo que pode ser utilizado para qualquer comentário que o usuário deseje fazer.

O resultado desta fase é um conjunto de formulários, que fornece a dimensão do número de nós da futura aplicação, apesar de não especificar completamente o seu conteúdo. A figura 4.14 mostra o diagrama em blocos para esta etapa:

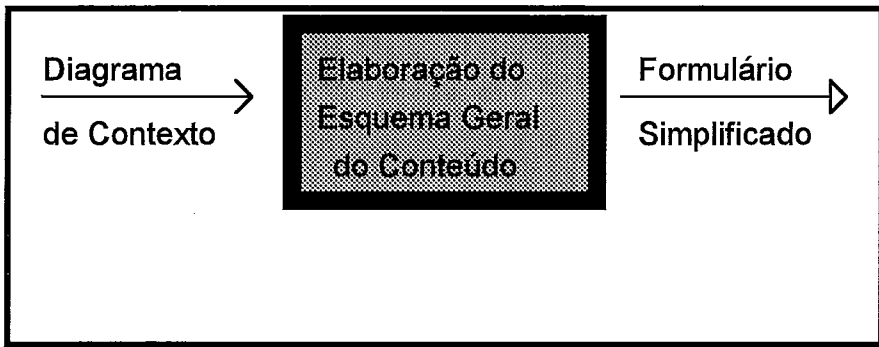


Figura 4.14 Diagrama em blocos para a etapa "Elaboração do Esquema Geral do Conteúdo"

4.4.3 Elaboração da rede de nós

Durante a etapa anterior a informação foi dividida em pequenas partições, chamadas nós. Cada um destes nós representa a "menor parte de um hiperdocumento que pode ser endereçada por uma ligação" [Ber 91]. Também podemos defini-los como "uma pequena porção do documento que cobre um determinado conceito" [Lit 91].

Desta maneira, o conjunto de Formulários Simplificados representa um esboço dos conceitos a serem abordados pela futura aplicação, separados em pequenas partes, que se relacionam entre si, através de ligações. Sob esta forma, a visualização da base de dados como um todo e das interligações que a compõem, fica bastante dificultada. Por outro lado esta visualização é fundamental para se poder avaliar se a organização da aplicação permite o alcance dos objetivos (requisitos adicionais) estabelecidos pelo produto.

De modo a solucionar este problema, o HiperAutor modela a base de dados a partir da informação contida nos formulários. Esta modelagem fornecerá uma representação gráfica dos nós englobados pela aplicação, bem como suas ligações.

Baseado nos formulários anteriores, resultado das várias interações com os especialistas do conteúdo, o analista prepara uma rede de nós onde ficam claras as diversas opções de caminho e os nós por onde há obrigatoriamente uma passagem. Para isso é de vital importância o registro dos objetivos definidos anteriormente. Para documentar a rede de nós tem-se um Formulário (figura 4.15) e uma linguagem específica (figura 4.16).

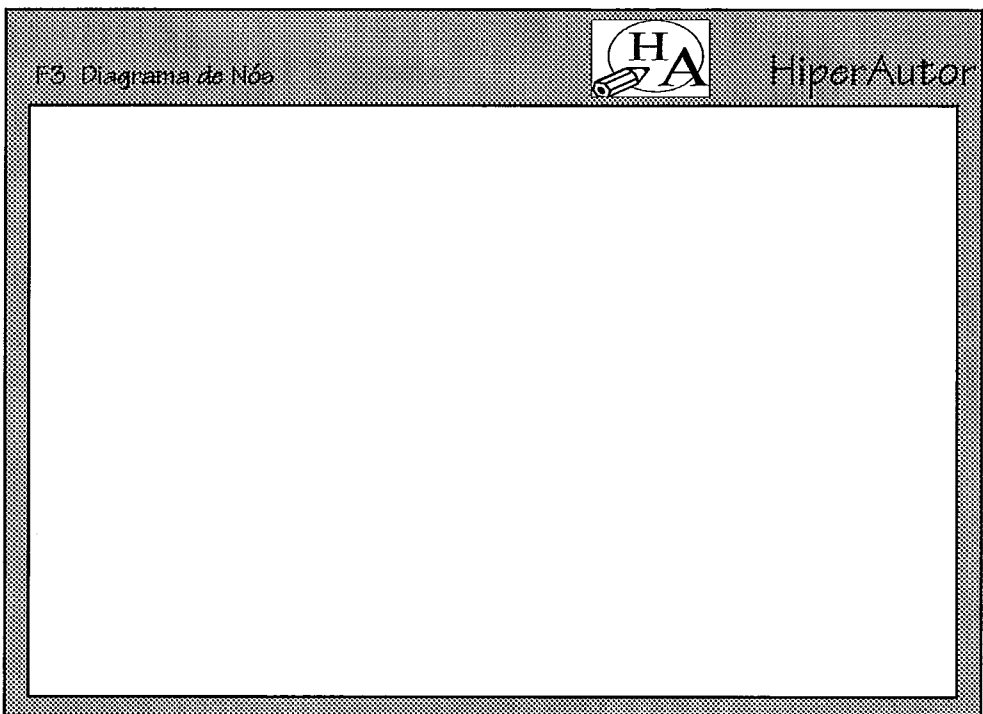


Figura 4.15 Formulário para Diagrama da Rede de Nós

Utilizando esta linguagem, é possível representar todos os nós e suas interligações, de modo a fornecer uma visão global do sistema.

A notação faz distinção entre três tipos de nós: nó simples, nó obrigatório e nó terminal. Esta classificação está relacionada ao conteúdo de cada nó e aos objetivos educacionais previamente definidos, permitindo que os autores estabeleçam nós obrigatórios, ou seja, aqueles em que o leitor necessariamente terá que passar, independentemente do caminho escolhido e, também, pontos de saída, que são aqueles onde o abandono do sistema é permitido pois não existe uma quebra de raciocínio.

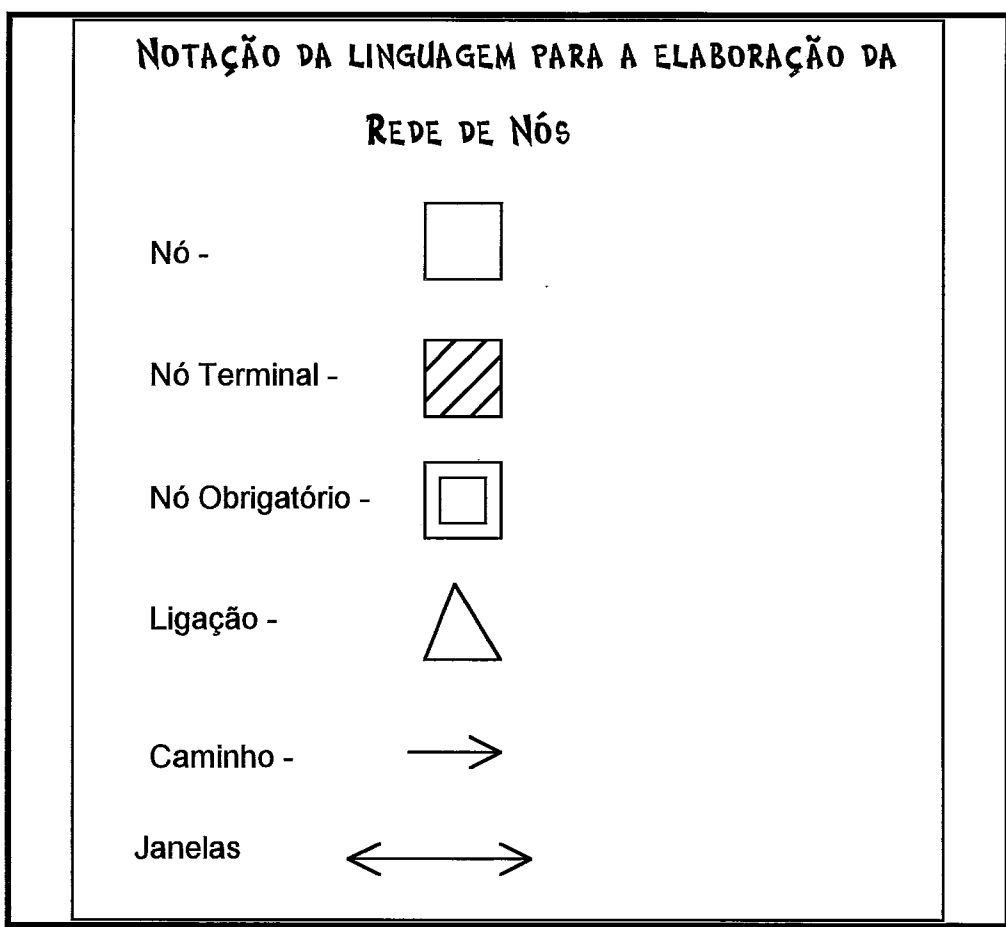


Figura 4.16 Notação da linguagem para a elaboração da Rede de Nós

Apesar desta opção parecer restritiva à liberdade de navegação, ela é que garante que o sistema favoreça o alcance dos objetivos definidos pelos especialistas do assunto. Além disso, a literatura tem mostrado

uma tendência que condena a execução de redes de nós interligados sem algum critério. DeYoung afirma que *"ligações arbitrárias tem sido identificadas como perigosas porque conduzem a produção de 'redes de nós do tipo Spaghetti', cujo efeito é muito parecido com o GOTO nas fases iniciais da programação"* [DeYo 90].

Atualmente, tem-se empregado a metáfora das "Excursões" [Hamm 88], que propõem um número de caminhos predeterminados, dentro de um corpo de informação. Desta forma, garantimos que o leitor terá acesso a todos os blocos fundamentais de informação, organizados de maneira clara, de modo a fornecer um ambiente de aprendizagem [Shne 91]. A linguagem definida para elaboração do Diagrama de Rede de Nós tem como objetivo suportar este conceito.

A figura 4.17 mostra o diagrama em blocos desta etapa.

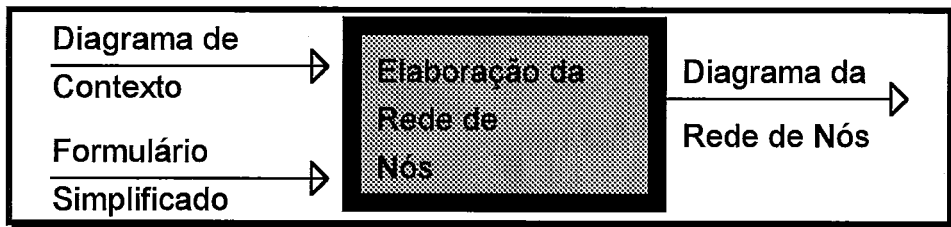


Figura 4.17 Diagrama em blocos da etapa "Elaboração da Rede de Nós"

4.4.4 Avaliação

Uma vez em posse do Diagrama da Rede de Nós, deve-se fazer uma avaliação do desenvolvimento do projeto. Para tal sugerimos o seguinte cenário:

- ◆ Os analistas de sistema, juntamente com os especialistas do assunto, fazem uma avaliação do produto como um todo, levando em conta todos os caminhos possíveis, a organização da informação e a complexidade tratada por cada um dos nós bem como a adequação dos nós obrigatórios e terminais tendo em conta os objetivos educacionais estabelecidos. Durante este processo, um dos analistas fica responsável por registrar as incorreções, incomplezas e inconsistências encontradas, bem como qualquer desejo de modificação expressa pelos especialistas no assunto. Este documento, chamado Laudo de Avaliação, será a base para a revisão do sistema, caso necessário.

A figura 4.18 mostra o diagrama em blocos para esta fase.

4.4.5 Especificação detalhada do projeto

Após a avaliação e aprovação da rede de nós pelos especialistas, é dado início à redação final do conteúdo dos nós. Devemos dar especial ênfase a esta tarefa, atuando de forma rigorosa, de maneira a evitar o excesso de informações em um único nó [Shei 91]. Wurman [Wur 89] afirma que a informação que não foi digerida representa nenhuma informação e somente cria no leitor a ilusão de que foi capaz de acessá-la, mas na realidade este não se beneficiou com a mesma. Desta forma, se não conferirmos uma organização adequada às informações da base de dados, estas serão perdidas.

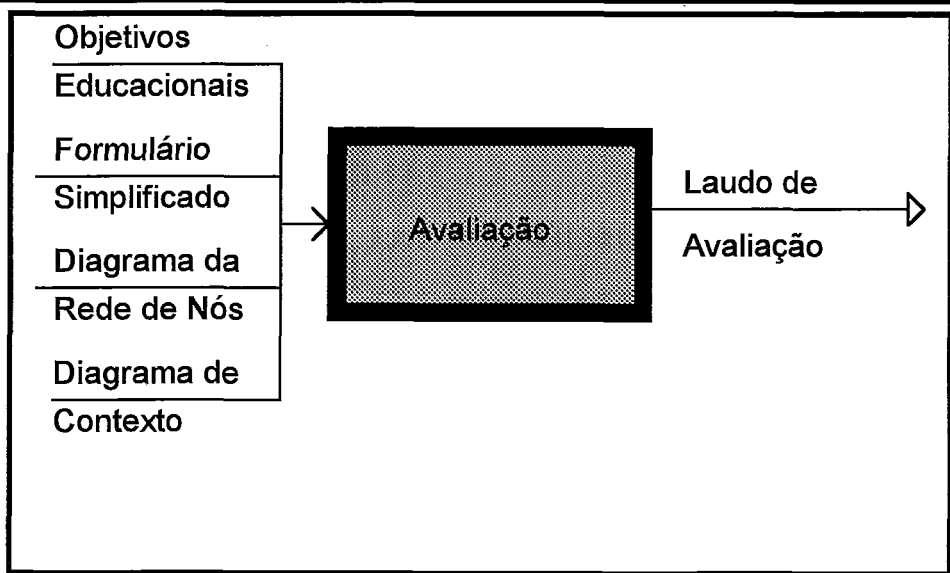




Figura 4.18 Diagrama em blocos para a etapa de Avaliação

Para minimizar este problema, Ben Shneiderman [Shn 91] propõe as seguintes técnicas para a redação de nós:

- ◆ Em vez de se discutir um tópico subsidiário, diferente do tópico principal do nó, deve-se somente fazer uma alusão ao mesmo e designar uma frase que se refira a ele através de uma ligação. Desta forma este tópico subsidiário se tornará assunto de um novo nó.
- ◆ A mesma técnica deve ser aplicada aos detalhes. Em vez de incluir a informação em um nó, deve-se fazer uma pequena referência conduzindo a um novo nó.

Levando em conta estas observações os especialistas no assunto, devem fazer a redação dos nós. Esta informação deve ser organizada no formulário 3 - Formulário Detalhado, como mostra a figura 4.19. Deve-se levar em conta a possibilidade de que neste processo de redação os especialistas optem por fazer a decomposição de um nó

F4 - Formulário Detalhado

HiperAutor



Texto

Alinhamento:
 Esquerda
 Centro
 Direita
 Justificado


Ligações (NomeBotão - NúmeroNóDestino)

1	5
2	6
3	7
4	8

Obs: /


Som

Tipo:
 Voz Humana
 Música


Imagem/Vídeo

Extensão:
 PCX
 BMP
 GIF
 AVI
 Outras:

Localização:
 Centro
 Esquerda
 Direita

Figura 4.19 Formulário detalhado

em dois ou mais nós de maneira a diminuir a quantidade de informação contida em cada um deles. A consequência deste

procedimento é uma revisão e alteração da Rede de Nós, de modo a registrar estas mudanças.

As normas para o preenchimento deste formulário são as seguintes:

- ◆ **Nome:** O nome é idêntico ao do Formulário Simplificado
- ◆ **Número:** O número é idêntico ao do Formulário Simplificado
- ◆ **Conteúdo:** Redação integral do texto do nó. Destacar as palavras/expressões que serão os botões através de letras maiúsculas.
- ◆ **Alinhamento:** Determinar o tipo de formatação do texto.
- ◆ **Ligações:** Indicar as os botões e o número dos nós aos quais remetem o usuário, quando acionado.
- ◆ **OBS:** Anotar qualquer comentário a respeito do texto
- ◆ **Som:** Descrição do texto ou música que se deseja digitalizado. Indicar a fonte e o interlocutor que se deseja para a gravação.
- ◆ **Nome do Arquivo:** Atribuir um nome para o arquivo de som, mesmo que este ainda não tenha sido digitalizado.
- ◆ **Imagem:** Descrição da imagem que se deseja inserir. Caso já exista indicar a fonte e se é necessário digitalizar. Se a imagem necessitar ser criada, fornecer os detalhes necessários para tal.
- ◆ **Nome do Arquivo:** Atribuir um nome para o arquivo de imagem, mesmo que este ainda não tenha sido digitalizado ou criado.
- ◆ **Localização:** Determinar a posição que se deseja colocar a figura, caso esta não ocupe a totalidade da tela.

É fundamental que o preenchimento deste formulário seja feito tendo em conta o Diagrama da Rede de Nós, pois o diagrama fornece a

cada momento uma visão global de todos os nós previstos e com conteúdo esboçados nos formulários especificados.

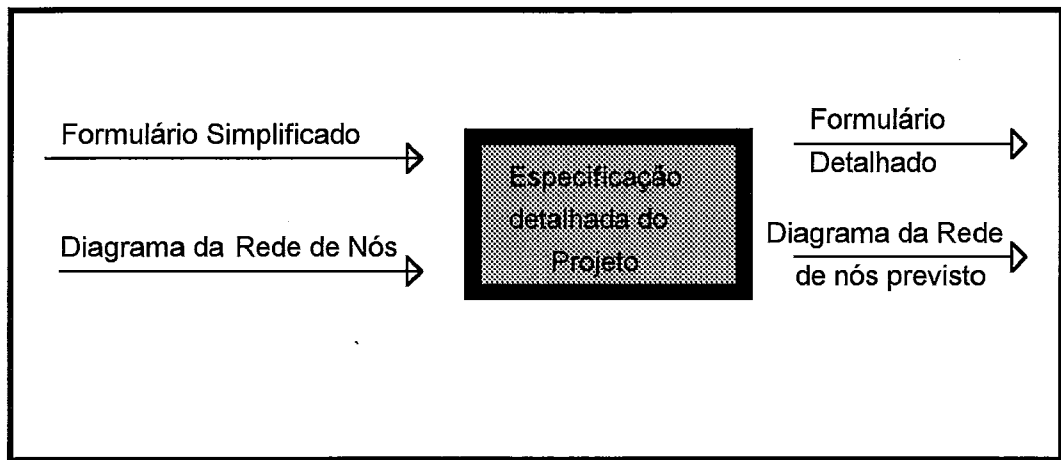


Figura 4.20 Diagrama em blocos da etapa "Especificação Detalhada do Projeto"

Uma outra função desta etapa é a escolha do software a ser usado na implementação. Critérios para a seleção deste tipo de software são descritos em [Mend 93].

4.4.6 Implementação da versão 1

O processo de especificação e projeto está terminado com a reunião do conjunto de formulários detalhados e a revisão do Diagrama da Rede de Nós, produzidos na fase de Especificação de Projeto e escolha do hardware e software a ser utilizado na implementação. O objetivo desta nova etapa é transformar as especificações de projeto em um produto operacional. Se as etapas anteriores forem realizadas de forma detalhada, o processo de implementação é mecânico e simples [Pres 92].

Acreditamos que a documentação gerada através da utilização do método proposto é suficiente para garantir a qualidade da implementação e, portanto, do produto final. É importante salientar o fato de que para esta etapa o ideal é ter-se implementadores que tenham familiaridade com o hardware e o software a serem utilizados na implementação. A figura 4.21 mostra o diagrama em blocos para esta fase.

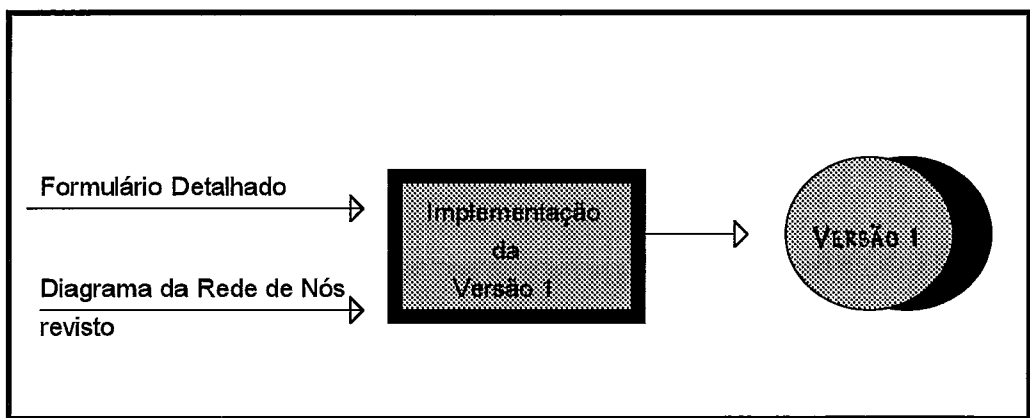


Figura 4.21 Diagrama em blocos da etapa "Implementação da Versão 1"

4.4.7 Avaliação

A avaliação do produto deve ser feita considerando-se os atributos de qualidade para hiperdocumentos descritos por Mendonça, [Mend 93]. Estes atributos estão sendo estudados e particularizados para aplicações educacionais de hipermídia por Campos (tese de mestrado em andamento). Os resultados iniciais estão descritos em [Camp 93].

4.4.8 Novas Versões

Nesta etapa será efetuada a manutenção/evolução do sistema, que definimos como sendo o conjunto de atividades que serão realizadas após o sistema ter sido entregue ao usuário [Ghez 91]. Basicamente

existirão dois tipos de manutenção: aquela que visa corrigir algum erro do sistema - manutenção corretiva - e a que acrescenta nós ao mesmo - manutenção evolutiva. Como acreditamos ser de fundamental importância uma documentação completa e atualizada do sistema, sugerimos que qualquer tipo de modificação a ser realizada, seja feita a partir do conjunto de formulários detalhados do Diagrama de Contexto e do Diagrama da Rede de Nós. O final desta etapa vai gerar uma nova versão desta documentação e do sistema, como é mostrado pelo diagrama em blocos da figura 4.22

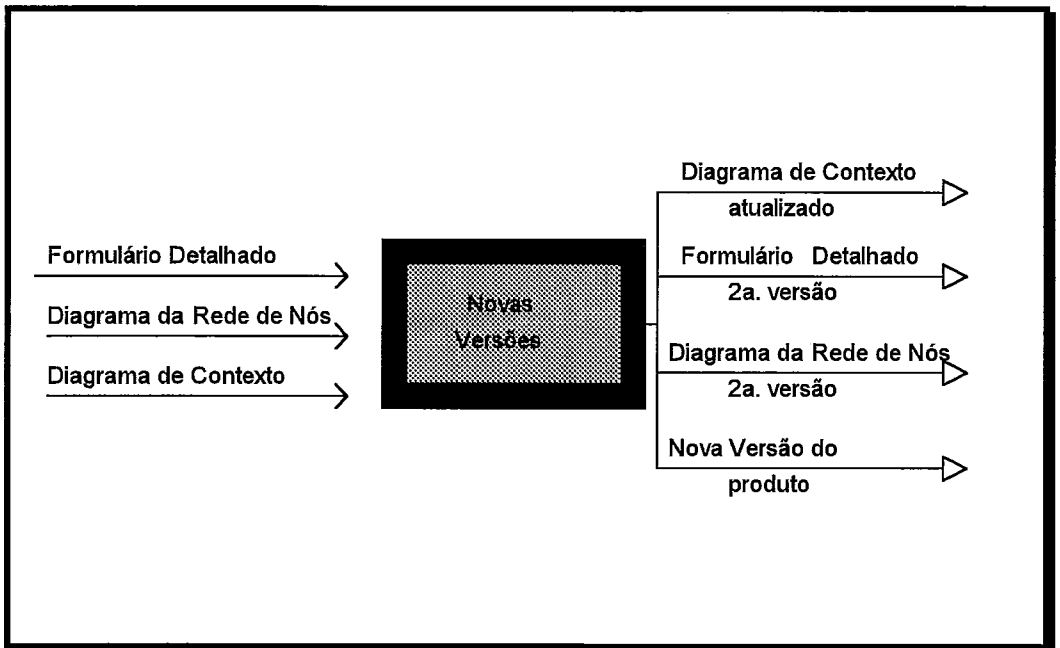


Figura 4.22 Diagrama em blocos da etapa "Novas Versões"

A literatura aponta esta como sendo a fase onde são gastos a maior parte dos recursos destinados ao desenvolvimento do software, na maior parte em consequência a falhas na especificação de requisitos [Pres 92, Ghez 91]. Acreditamos que a utilização do método proposto venha a minimizar esta problemática.

CAPÍTULO 5



Conclusão

Esta tese apresentou o Hiper Autor, um método para especificação e projeto de aplicações educacionais do tipo hipertexto/hipermídia que estabelece uma forma de organizar e disciplinar o pensamento e o trabalho de uma equipe desenvolvedora, composta por profissionais de várias áreas, estabelecendo para cada etapa no processo de desenvolvimento da aplicação técnicas e instrumentos adequados para sua realização.

O método, apoiado no modelo de ciclo de vida em espiral, prevê uma série de interações entre a equipe ao longo do processo de desenvolvimento. Estas interações são registradas através de uma linguagem gráfico-linguística, representada por um conjunto de formulários, que uma vez preenchidos, fornecerão a documentação do sistema. Esta documentação, além de facilitar a manutenção do sistema, permite que a qualidade do sistema seja avaliada em várias etapas do seu desenvolvimento.

O método apresentado responde a uma necessidade de organizar o processo de autoria de aplicações do tipo hipertexto/hipermídia, identificado tanto na literatura quanto junto a grupos desenvolvedores com pouca experiência em Informática. Tanto a autoria em ponto grande quanto em ponto pequeno são contempladas, de modo a apoiar integralmente a equipe desenvolvedora. O método é independente da ferramenta a ser utilizada para implementação da aplicação, de maneira a permitir a elaboração de aplicações educacionais do tipo hipertexto/hipermídia para qualquer plataforma.

O tema abordado fornece várias possibilidades para futuros trabalhos e pesquisas. Além da implementação de ferramentas de apoio do Hiper Autor, vislumbramos a possibilidade de aliar o método proposto a técnicas de reutilização, no sentido de permitir o reuso de especificações de aplicações já existentes na construção de novas aplicações. As especificações reutilizáveis serão geradas através do Hiper Autor.

Uma outra possibilidade é a extensão e aprofundamento dos critérios para a avaliação da qualidade de aplicações do tipo hipertexto/hipermídia. Este assunto já é objeto de uma tese de mestrado, em andamento na COPPE/UFRJ.

Finalmente, podemos considerar uma possível atuação na área de pesquisa em interface com o usuário, uma vez que o Hiper Autor, em suas etapas finais, especifica a interface do sistema (cor de fundo, botões, posicionamento de figuras na tela ...). Esta é uma área que

tem ganhado cada vez mais importância a partir da evolução das interfaces gráficas e é fator determinante para o sucesso de qualquer aplicação. O surgimento de novas tecnologias de interfaces, tais como a realidade virtual, demandam estudos no sentido de se determinar métodos mais adequados de especificação e projeto de sistemas hipertexto/hipermídia. Apontamos a possibilidade de se estender o método Hiper Autor de maneira a atender esta problemática.

Referências Bibliográficas

- ◆ [Agui 92] - Aguiar, Teresa Cristina; Um sistema especialista de suporte decisão para planejamentos de ambientes de desenvolvimento de software; Tese de doutorado; COPPE/ UFRJ; 1992.
- ◆ [Aksc 88] - Akscyn, Robert et alii; KMS a distributed hypermedia system for managing knowledge in organizations; Communications of the ACM July 88 Volume 31; 1988.
- ◆ [Alty 93] - Alty, James; Multimedia: We have the tecnologia but do we have a methodology ?; Proceedings of ED MEDIA 93; Educational Multimedia and Hypermedia Annual; AACE - 1993.
- ◆ [Barb 85] - Barbie, F.; The type concept in office document retrieval; Proceedings of the conference in very large databases; 1985.
- ◆ [Berk 91a] - What is Hypertext?; Hypertext and Hypermedia Handbook; Mc Graw Hill; 1991.
- ◆ [Berk 91b] - A hipertext Glossary; Hypertext and Hypermedia Handbook ; Mc Graw Hill; 1991.
- ◆ [Bers 91] -Bersoff, E. e Davis, A.; Impacts of life cycle models on software; Communications of the ACM -Agosto; 1991.
- ◆ [Boar 84] - Application prototyping; Journal of systems management; Fevereiro; 1984.

- ◆ [Boeh 88] Boehm, B. ; A spiral model of software development Computer; maio; 1988.
 - ◆ [Camp 88] - Campbell, B. e Goodman, J.M.; HAM : A general purpose hypertext abstract machine; Communications of the ACM 31,7 Julho; pp. 856-861; 1988.
 - ◆ [Camp 92] - Campos, F. C.A. et alii; Autorialia e trabalho cooperativo, reflexões sobre uma experiência em hipertexto; Workshop em Trabalho Cooperativo, Pensamento crítico e software educacional; COPPE/UFRJ; 27 de Agosto de 1992.
 - ◆ [Char 86] - Charette, R.; Software Engineering Environments: Concepts and Technology; Mc Graw Hill; New York; 1986.
 - ◆ [Chen 76] - Chen, P. P.; The Entity Relationship Model: Toward a unified view of data; ACM transactions on Data Base Systems; v. 1(1), 1976.
 - ◆ [Conk 87] -Conklin, J.; gIBIS : a hypertext tool for team design deliberation; Proceedings on Hypertext '87; 1987.
 - ◆ [Davi 90] - Davis. Alan M. ; Software Requirements, analysis & specification ; Prentice Hall international editions ; 1990
 - ◆ [Davi 92] - Davis. Alan M. ; Operational prototyping: a new development approach; Communications of the ACM; setembro; 199
 - ◆ [DeYo 88] - DeYoung, L. ; Linking considered harmful. ; Ritz, A. , Streitz, N. , & André , J. , editors Hipertexts: Concepts, Systems and applications, Proc ECHT '90 ; Cambridge University Press; 1988
 - ◆ [Fair 85] - Fairley, R.E. ; Software Engineering Concepts;McGraw Hill, 1985.
 - ◆ [Fide 88] - Fiderio, Janet - a grand Vision - Byte outubro; 1988
-

- ◆ [Gane 79] -Gane,Chris and Sarson,Trish; Structured Systems Analysis; Prentice Hall ; 1979.
 - ◆ [Garz 90] - Garzotto, Franca; Paolini, Paulo; Schwabe, Daniel ; HDM a model based approach to hypertext design ;Politecnico de Milano;Italy 1990.
 - ◆ [Garz 91] - Garzotto, Franca; Paolini, Paolo; Schawabe, Daniel ; Tools for designing Hyperdocuments; Hypertext and Hypermedia Handbook; Mc Graw Hill 1991.
 - ◆ [Gay 91] - Geri Gay - Structuring interactive multimedia fiction; Hypertext and Hypermedia Handbook; Mc Graw Hill 1991.
 - ◆ [Ghez 91] - Ghezzi, C.; Jazayeri, M. ; Mandrioli, D. ; Fundamentals of software engineering; Prentice Hall International; 1991
 - ◆ [Glush 89] - Glusko, Robert; "Hypertext 89 course notes #3: Turning Text into Hypertext; 1989
 - ◆ [Hold 92] - Holden, M. C. et alii ; J.F. Designing HyperCard Stacks for Cooperative learning. The computing teacher Feb 1992,20-25
 - ◆ [Hora 85] - Horak, W. ; Office document architecture and office document interchange formats: current status of international standardization ; IEEE Computer Oct. ; 1985
 - ◆ [Jona 88] -Jonassen, D. Designing Structured Hypertext and Structuring. *Educational Technology*, Novembro, 1988.
 - ◆ [Kyng 91] - Kyng, Morten - Designing for cooperation - ACM - v.34 ; 1991.
 - ◆ [Leit 91] - Leite,J.; Validação de requisitos: O uso de pontos de Vista ; Revista Brasileira de Computação ; OUT/DEZ 1990 V.6 N.2 Pag 39;52; 1991.
-

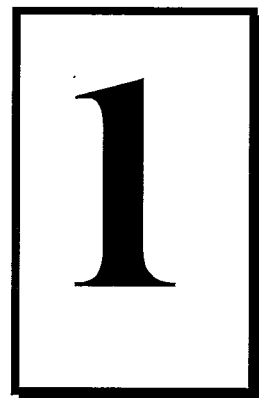
- ◆ [Maye 93] Mayer, Stefan et alii - NEAT: An integrated Authoring Environment Based Upon Toolbook; Proceedings of ED MEDIA 93; Educational Multimedia and Hypermedia Annual; AACE ; 1993.
 - ◆ [Mend 92] - Mendes, Maria Emília Xavier; Uma paradigma para hipertexto e o processo de ensino/aprendizagem: uma relação promissora; Tese de Mestrado; COPPE/ UFRJ; 1992.
 - ◆ [Mend 93] - Mendonça, Luciana F. e Rocha A. R. ; Critérios de qualidade para avaliação de sistemas de hipertexto ; Encontro França Brasil de Informática na Educação ; Rio de Janeiro 1993.
 - ◆ [Mull 79] - Mullery, G.; CORE: a method for controlled requirement specification. ; In Proc. 4th international conference on software engineering IEEE computer society press pp126;135; 1979.
 - ◆ [Niel 90] - Nielsen, Jakob ; Hypertext and Hypermedia; Academic Press INC. ; 1990.
 - ◆ [Pere 91] - Perez, Ernest ; Tools for authoring Hypertexts; Hypertext and Hypermedia Handbook ; Mc Graw Hill; 1991.
 - ◆ [Petr 93] - Petrushin, Valery A. ; HELENA Hypermedia Project: from knowledge representation to knowledge construction ; Proceedings of ED MEDIA 93; Educational Multimedia and Hypermedia Annual; AACE ; 1993.
 - ◆ [Plais 91] - Plaisant, C.; Guide to Opportunities in Volunteer Archeology: Case Study on the use of a Hypertext System in a museum Exhibit ; Hypertext and Hypermedia Handbook ; Mc Graw Hill; 1991.
 - ◆ [Press 92] - Pressman, Roger; Software Engineering: a practitioner's approach; Third edition; Mc Graw Hill International editions; 1992
-

- ◆ [Roch 87] - Rocha, A.R. et alii ; Ambiente para desenvolvimento de software: Definição de Termos; Relatório Técnico ES;137/87; COPPE/UFRJ; 1987.
 - ◆ [Roch 92] - Rocha, A.R. et alii ; Esperiências no desenvolvimento de software educacional; III Simpósio Brasileiro de Informática na Educação; Rio de Janeiro; 1992.
 - ◆ [Ross 75] - Ross, D. ; structured analysis for Requirements definition ; IEEE Trans. Software engineering, vol 3 n. 1 January 1975 pp. 5;16; 1975.
 - ◆ [Salg 92] -Salgado, A.; Fonseca, D.; Albuquerque, E.; Meira, S. Sistemas Hipermídia: Hipertexto e Bancos de Dados. Porto Alegre: Instituto de Informática da UFRGS, 1992.
 - ◆ [Sequ 92] - Sequerra, K. - Hiper Autor: uma ferramenta para o trabalho cooperativo - Workshop em trabalho cooperativo, pensamento crítico e software educacional - COPPE/UFRJ; 1992
 - ◆ [Shne 89] - Shneiderman, Ben et alii ; Hypertext Hands On! ; Addison Wesley Massachusetts ; 1989
 - ◆ [Shne 91] - Shneiderman, Ben et alii ; Editing to Structure a Reader's Experience ; Hypertext and Hypermedia Handbook ; Mc Graw Hill; 1991.
 - ◆ [Slat 91] - Slatin, John; Composing hypertext: a discussion for writing teachers; Hypertext and Hypermedia Handbook ; Mc Graw Hill; 1991.
 - ◆ [Sobi 91] - Sobiesak, Rick ; Mylopoulos, John ; A conceptual Modelling approach to authoring;in-the-large for Hypertext documents; ACM; 1991
-

- ◆ [Wurm 89] ; Wurman, R.S. ; *Information Anxiety* ; DoubleDay, New York; 1989.
 - ◆ [Your 89] ; Yourdon, E. ; *Análise Estruturada de Sistemas* ; Editora Campus; 1990.
 - ◆ [Your 92] - Yourdon, E.; *Decline and fall of the american programmer*; Yourdon Press ; 1992.
-

ANEXO 1

Fomulários do Hiper Autor





Objetivos:



Nome do Nó: _____

Número: _____

Contexto: _____

 Texto

Conteúdo

Ligações (NomeBotão - NúmeroNóDestino)

- | | |
|-----|-----|
| A - | E - |
| B - | F - |
| C - | G - |
| D - | H - |

Obs:!

 Som

Descrição:

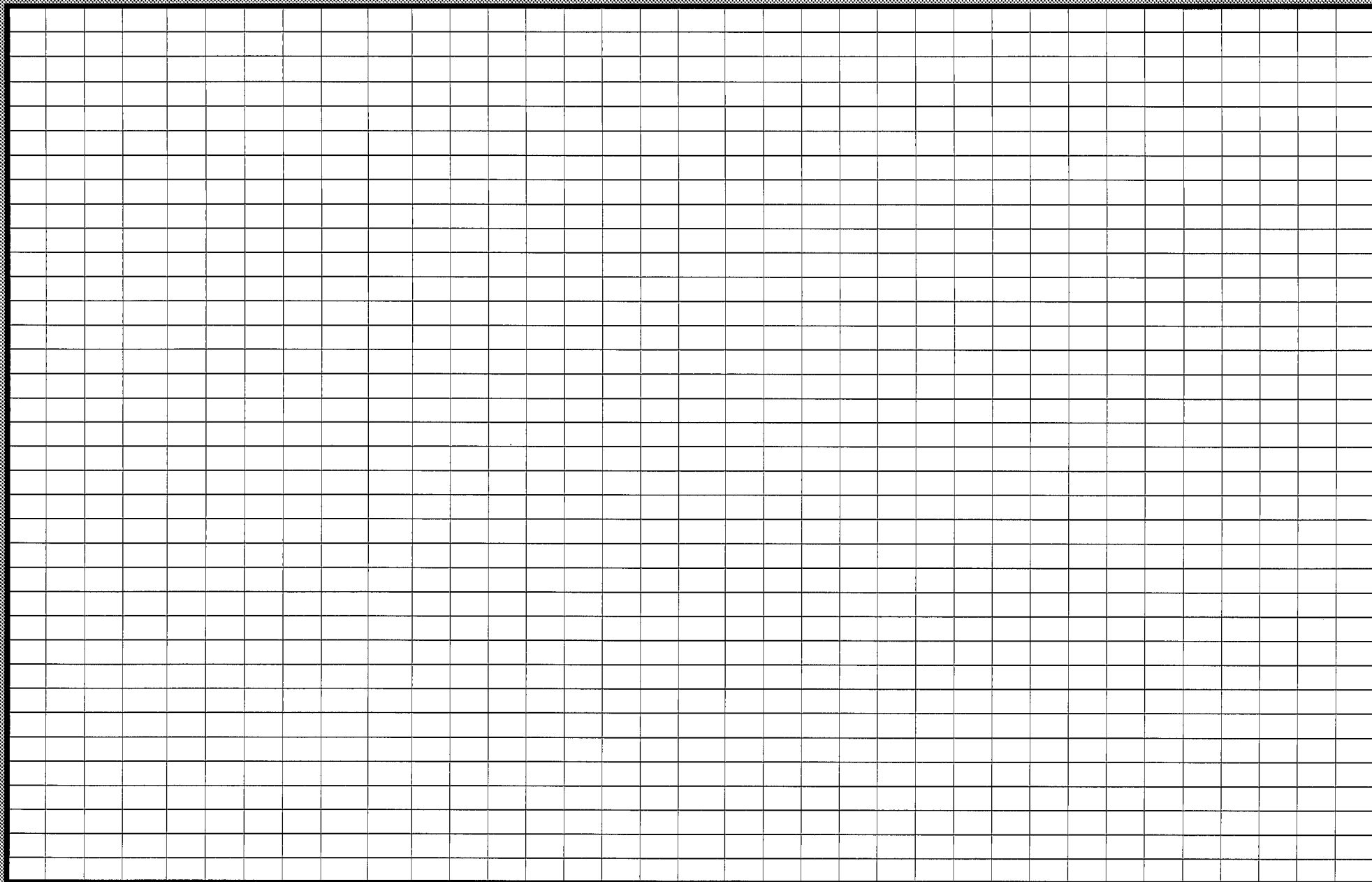
 Imagem/Vídeo

Descrição:

F3 Diagrama de Nós



HiperAutor





Nome do Nó:	Número:
Contexto:	
Background:	

 Texto

Conteúdo	

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1	5
2	6
3	7
4	8

Obs:!

 Som

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:

 Imagem/Vídeo

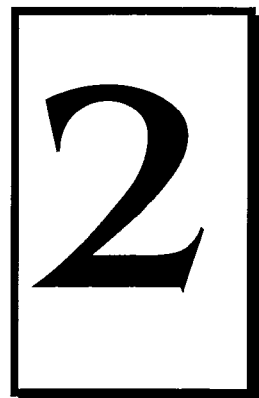
Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo:

Localização: Centro Esquerda Direita

ANEXO 2

Exemplo de Utilização do Hiper Autor



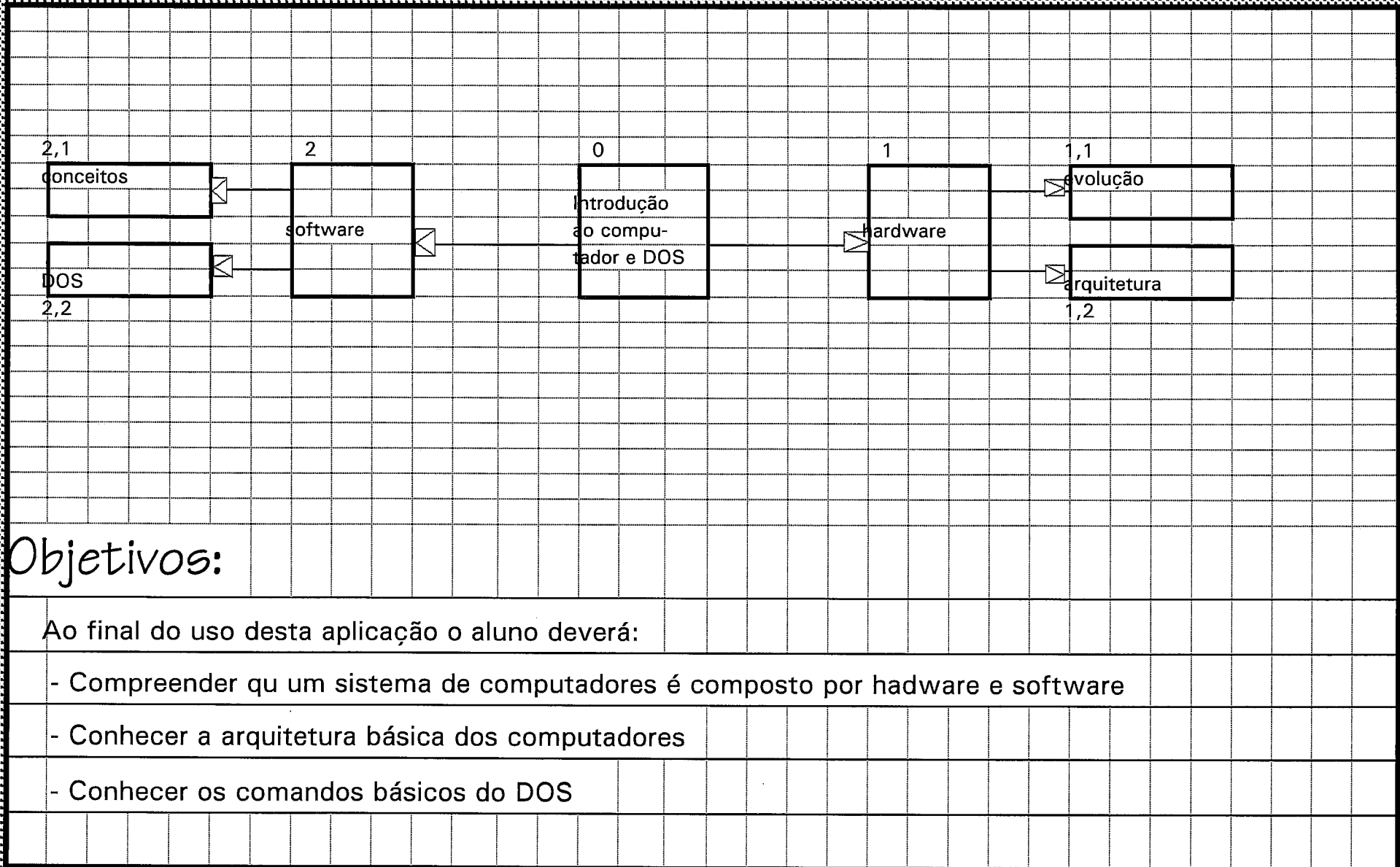
A seguir apresentaremos um exemplo de aplicação do método Hiper Autor. O tema escolhido é uma aplicação instrucional do uso de computadores onde são introduzidos conceitos básicos do sistema operacional DOS. O exemplo se divide da seguinte forma:

- ◆ O primeiro formulário - Diagrama de Contexto - mostra uma visão geral da aplicação, apresentando a divisão em contextos e subcontextos.
 - ◆ O conjunto de formulários simplificados que vêm a seguir, representa a divisão do contexto de Hardware, previsto no formulário anterior, em nós e elos. Para efeito deste exemplo, optamos por somente apresentar a elaboração deste contexto, que julgamos suficiente para a compreensão do método proposto.
 - ◆ À partir do conjunto de formulários simplificados foi possível construir a rede de nós da aplicação. Esta representação se encontra disponível no formulário de diagrama da rede de nós.
 - ◆ Finalmente apresentamos o conjunto de formulários detalhados, que representa a elaboração da redação e aspecto visual dos nós previstos anteriormente. É importante frisar que o conjunto destes formulários, juntamente com o diagrama da rede de nós, fornecem a especificação de requisitos para a implementação da aplicação.
-

F1 Diagrama de Contexto



HiperAutor



Objetivos:

Ao final do uso desta aplicação o aluno deverá:

- Compreender qu um sistema de computadores é composto por hadware e software
- Conhecer a arquitetura básica dos computadores
- Conhecer os comandos básicos do DOS



Nome do Nó: Cover Número: i

Contexto: Introdução a Dos

Texto

Conteúdo
Título do trabalho + ícone de HELP

Ligações (NomeBotão - NúmeroNóDestino)

A - Help

E -

B -

F -

C -

G -

D -

H -

Obs:

Som

Descrição:

Imagem/Vídeo

Descrição:
Fotos de computadores



Nome do Nó: Help

Número: ii

Contexto: Introdução a DOS



Texto

Conteúdo

Apresentar a forma de navegação básica

Introduzir os ícones de : VOLTA

SAÍDA

Ligações (NomeBotão - NúmeroNóDestino)

- A - E -
- B - F -
- C - G -
- D - H -

Obs:

JANELA



Som

Descrição:




Imagem/Vídeo

Descrição:




Nome do Nó: Tela Introdutória Número: 0
 Contexto: Introdução a DOS


Conteúdo	 Texto
	Para que servem os computadores
	Dividem-se em HARDWARE
	SOFTWARE

Ligações (NomeBotão - NúmeroNóDestino)

A- HARWARE	1	E-
B- SOFTWARE	2	F-
C-		G-
D-		H-

Obs.:

 Som
Descrição:

 Imagem/Vídeo
Descrição:



Nome do Nó: HARDWARE

Número: 1,0

Contexto: HARDWARE



Texto

Conteúdo

Evolução do Hardware nas últimas décadas

Arquitetura básica dos computadores

Ligações (NomeBotão - NúmeroNóDestino)

- A - EVOLUÇÃO 1,1 E -
- B - ARQUITETURA BÁSICA 1,2 F -
- C - G -
- D - H -

Obs:



Som

Descrição:



Imagem/Vídeo

Descrição:

Fotos comparando o tamanho do hardware de várias máquinas



Nome do Nó: Evolução

Número: 1,1

Contexto: HARWARE



Texto

Conteúdo

Histórico da evolução do Hardware

Apresenta Mainframe

Aparecimento dos microcomputadores

Ligações (NomeBotão - NúmeroNóDestino)

A - MAINFRAME 1,3 E -

B - MICROCOMPUTADORES 1,4 F -

C - G -

D - H -

Obs: /



Som

Descrição:



Imagem/Vídeo

Descrição:



Nome do Nó: ARQUITETURA BÁSICA Número: 1,2
 Contexto: HARDWARE

 Texto

Conteúdo

Apesar da grande variedade de modelos existentes no mercado, a arquitetura básica dos computadores é sempre composta por:


MEMÓRIA	DISPOSITIVOS DE ENTRADA/SAÍDA
PROCESSADOR	DISPOSITIVOS DE ARMAZENAMENTO

Remeter para o contexto de Software

Ligações (NomeBotão - NúmeroNóDestino)

- | | |
|---------------------------|--------------------|
| A- Memória 1,2,1 | E- Software 2,0 |
| B- Processador 1,2,2 | F- |
| C- Entrada/Saída 1,2,3 | G- |
| D- Armazenamento 1,2,4 | H- |

Obs:

 Som

Descrição

 Imagem/Vídeo

Descrição:

Figura com a arquitetura básica dos computadores.



Nome do Nó: MEMÓRIA

Número: 1,2,1

Contexto: HARDWARE



Texto

Conteúdo

Breve descrição

Ligações (NomeBotão - NúmeroNóDestino)

- | | |
|-----|-----|
| A - | E - |
| B - | F - |
| C - | G - |
| D - | H - |

Obs:

JANELA



Som

Descrição:



Imagem/Vídeo

Descrição:



Nome do Nó: PROCESSADOR

Número: 1,2,2

Contexto: HARDWARE



Texto

Conteúdo

Breve descrição da linha INTEL (286,386, ..)

Ligações (NomeBotão - NúmeroNóDestino)

- | | |
|-----|-----|
| A - | E - |
| B - | F - |
| C - | G - |
| D - | H - |

Obs.:

JANELA



Som

Descrição:



Imagem/Vídeo

Descrição:

FOTOS DOS PROCESSADORES INTEL



Nome do Nó: DISPOSITIVOS de E/S

Número: 1,2,3

Contexto: HARDWARE



Texto

Conteúdo

Breve descrição dos dispositivos mais usuais: teclado, mouse...

Ligações (NomeBotão - NúmeroNóDestino)

A -

E -

B -

F -

C -

G -

D -

H -

Obs:

JANELA



Som

Descrição:




Imagem/Vídeo

Descrição:



Nome do Nó: DISP. de Armazenamento Número: 1,2,4
Contexto: HARDWARE


Conteúdo  Texto

Breve descrição


Ligações (NomeBotão - NúmeroNóDestino)

A -	E -
B -	F -
C -	G -
D -	H -

Obs: JANELA

 Som

Descrição:

 Imagem/Vídeo

Descrição:



Nome do Nó: mainframes Número: 1,3
 Contexto: HARDWARE

A Texto

Conteúdo: O que são mainframes?

Supercomputadores

Arquitetura dos mainframes

Ligações (NomeBotão - NúmeroNóDestino)

- | | | |
|-----------------------|-------|-----|
| A - supercomputadores | 1,3,1 | E - |
| B - arq. básica | 1,2 | F - |
| C - | | G - |
| D - | | H - |

Obs.:

M Som

Descrição:

V Imagem/Vídeo

Descrição:



Nome do Nó: SUPERCOMPUTADORES Número: 1,3,1

Contexto: HARDWARE



Texto

Conteúdo

Descrição de um Supercomputador

Ligações (NomeBotão - NúmeroNóDestino)

- | | |
|-----|-----|
| A - | E - |
| B - | F - |
| C - | G - |
| D - | H - |

Obs.:

JANELA



Som

Descrição:



Imagem/Vídeo

Descrição:

FOTO DE UM SUPERCOMPUTADOR



Nome do Nó: MICROCOMPUTADORES Número: 1,4
 Contexto: HARDWARE



Texto

Conteúdo

Breve histórico da computação

Linha PC

Arquitetura básica dos PC's

Ligações (NomeBotão - NúmeroNóDestino)

- | | | |
|-----------------|-----|-----|
| A - PC's | 1,4 | E - |
| B - ARQUITETURA | 1,2 | F - |
| C - | | G - |
| D - | | H - |

Obs: /



Som

Descrição:



Imagem/Vídeo

Descrição:



Nome do Nó: PC's Número: 1,4,1
Contexto: HARDWARE

 Texto

Conteúdo

Histórico dos PC's

Ligações (NomeBotão - NúmeroNóDestino)

- | | |
|-----|-----|
| A - | E - |
| B - | F - |
| C - | G - |
| D - | H - |

Obs.:

JANELA

 Som

Descrição:

 Imagem/Vídeo

Descrição:

FOTO DE UM PC



Nome do Nó: SOFTWARE Número: 2,0
 Contexto: SOFTWARE

 Texto

Conteúdo
 O que é software?

Vários tipos de software: APLICATIVOS

SISTEMAS OPERACIONAIS

LINGUAGENS DE PROGRAMAÇÃO

Volta para o contexto de Hardware

Ligações (NomeBotão - NúmeroNóDestino)

A - APLICATIVOS	2,02 E -
B - SIST. OPERACIONAIS	2,1 F -
C - LINGUAGENS	2,01 G -
D - HARWARE	1,0 H -

Obs:

 Som

Descrição:

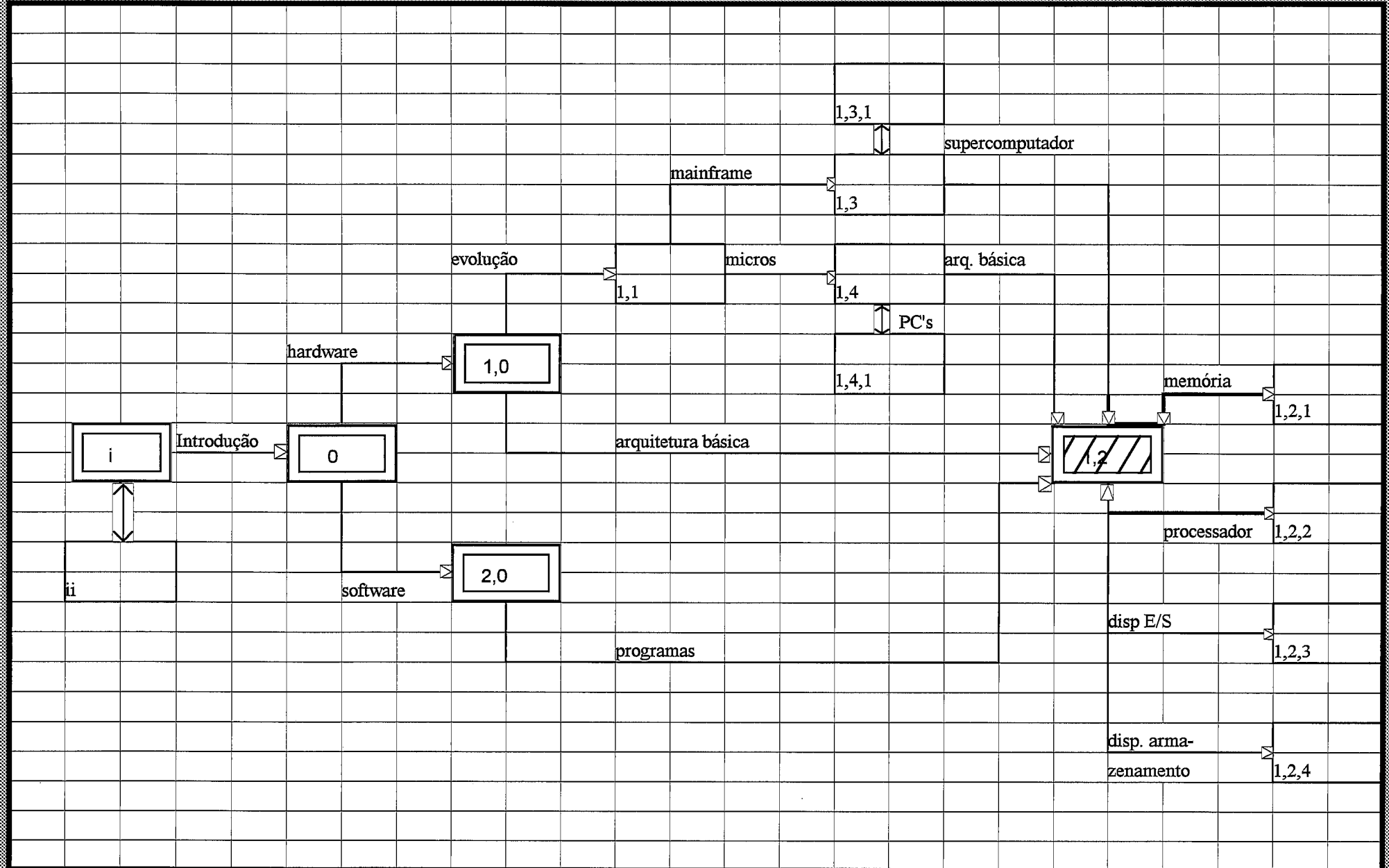
 Imagem/Vídeo

Descrição:

F3 Diagrama de Nós



HiperAutor



Nome, do Nó: COVER Número: i
 Contexto: INTRODUÇÃO A DOS
 Background: COVER

 Texto

Conteúdo

TÍTULO: INTRODUÇÃO AO COMPUTADOR COM DOS

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

- | | |
|---|---|
| 1 HELP - ii | 5 |
| 2 qualquer ponto da tela remete ao prefácio | 6 |
| 3 | 7 |
| 4 | 8 |

Obs: /

 Som

Tipo: Voz Humana Música

Nome do Arquivo: FUNDO.MIDI

Descrição:

MÚSICA DE FUNDO - BACH Concertos de Brandenburgo- 3

 Imagem/Vídeo

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo: PC_COVER.PCX

Localização: Centro Esquerda Direita



Nome do Nó: HELP

Número: ii

Contexto: INTRODUÇÃO A DOS

Background: PADRÃO



Texto

Conteúdo

Para se dirigir a qualquer um dos tópicos que estiverem em destaque na tela, basta clicar uma única vez sobre o mesmo.

Caso nenhum tópico estiver em destaque, clique sobre o ícone V. Você voltará a tela anterior.

Se desejar abandonar o sistema, clique sobre o ícone S, quando este estiver disponível.

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NomeNóDestino)

1	5
2	6
3	7
4	8

Obs.:

JANELA



Som

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:




Imagem/Vídeo

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo:

Localização: Centro Esquerda Direita

Nome do Nó: TELA INTRODUTÓRIA	Número: 0
Contexto: INTRODUÇÃO A DOS	
Background: PADRÃO	



Texto


Conteúdo	Com o auxílio do computador você pode produzir cartas e relatórios, desenhar um mapa de cidade, fazer um balanço de sua bancária, criar gráficos, pesquisar preços de estoque, jogar e realizar tarefas.
	Para tanto é necessário saber conceitos básicos de :
	HARWARE
	SOFTWARE

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1 HARWARE	1		5
2 SOFTWARE	2		6
3			7
4			8

Obs: /




Som

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:



Imagem/Vídeo

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo:

Localização: Centro Esquerda Direita



Nome do Nó: HARDWARE	Número: 1,0
Contexto: HARDWARE	
Background: PADRÃO	

 **Texto**

Conteúdo	Harware é a máquina propriamente dita. A evolução do hardware nos últimos 40 anos tem sido surpreendente. Dos computadores mainframes, aos microcomputadores temos hoje computadores cada vez mais poderosos e velozes, que tem uma mesma arquitetura básica.

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1	<u>EVOLUÇÃO</u>	1,1	5
2	<u>ARQ. BÁSICA</u>	1.2	6
3			7
4			8

Obs:!

 **Som**

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:

 **Imagem/Vídeo**

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo: **HW_BMP E HW_93.BMP**

Localização: Centro Esquerda Direita

Nome do Nó: EVOLUÇÃO	Número: 1,1
Contexto: HARDWARE	
Background: PADRÃO	

 **Texto**


Conteúdo	A evolução dos computadores tem sido crescente. O primeiro computador, o ENIAC, construído em 1946, funcionava com 18000 válvulas e pesava 70 toneladas.
	O desenvolvimento do transistor nos anos 50 permitiu a produção de computadores menores e eficientes. Na década de 60, a invenção dos circuitos integrados possibilitou o surgimento de máquinas microprogramadas. Nos anos 80, as tecnologias de encapsulamento permitiram grandes avanços na microeletrônica, permitindo o surgimento dos microcomputadores.

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1 <u>M</u> ainframes	1,3	5
2 <u>M</u> icrocomputadores	1,4	6
3		7
4		8

Obs: /

 **Som**

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:

 **Imagem/Vídeo**

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo:

Localização: Centro Esquerda Direita

Nome do Nó: ARQUITETURA BÁSICA	Número: 1,2
Contexto: HARDWARE	
Background: PADRÃO	

 **Texto**

Conteúdo

Apesar do grande número de modelos de computadores existentes atualmente no mercado tanto os computadores de grande porte - Mainframes- quanto os microcomputadores, possuem a mesma arquitetura básica.

Podemos dividir a arquitetura básica de um computador em quatro blocos:
MEMÓRIA, PROCESSADOR, DISPOSITIVOS DE ENTRADA E SAÍDA E DISPOSITIVOS DE ARMAZENAMENTO. Para que o computador funcione, são necessários PROGRAMAS.

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

A	Memória	1,2,1	E	Programas	2,0
B	Processador	1,2,1	F		
C	Dispositivos de Ent/Saída	1,2,3	G		
D	Dispositivos de Armazenamento	1,2,4	H		

Obs.: Memória, processador e dispositivos são janelas. Programas remete para o contexto de software

 **Som**

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:

 **Imagem/Vídeo**

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo: **ARQ_PC.PCX**

Localização: Centro Esquerda Direita



Nome do Nó: MEMÓRIA	Número: 1,2,1
Contexto: HARDWARE	
Background: PADRÃO	

A Texto

Conteúdo	É a área de trabalho dos computadores, também chamada de RAM - Random Access
----------	--

Memory	

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1	5
2	6
3	7
4	8

Obs: /

JANELA

Som

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição

Imagem/Vídeo

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo: RAM.PCX

Localização: Centro Esquerda Direita

Nome do Nó: PROCESSADOR	Número: 1,2,2
Contexto: HARDWARE	
Background: PADRÃO	

 **Texto**

Conteúdo	No processador são realizadas as operações lógicas e aritméticas do computador.
----------	---

Os processadores mais conhecidos são os fabricados pela INTEL: 286, 386, 486 e Pentium

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroINóDestino)

1	5
2	6
3	7
4	8

Obs:!

JANELA

 **Som**

Tipo: Voz Humana Música

Nome do Arquivo:

Descriç

 **Imagem/Vídeo**

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo: 286.pcx 386.pcx e 386.pcx

Localização: Centro Esquerda Direita

Nome do Nó: Dispositivos de entrada e saída	Número: 1,2,3
Contexto: HARDWARE	
Background: PADRÃO	

 **Texto**

Conteúdo	Os dispositivos de entrada e saída permitem inicializar dados no computador e obter registros de um processamento. Os mais conhecidos são : Teclado, mouse, impressora e monitor.

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1	5
2	6
3	7
4	8

Obs: JANELA

 **Som**

Tipo: Voz Humana Música

Nome do Arquivo:
 Descrição:

 **Imagem/Vídeo**

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo: **teclado.pcx; mouse.pcx e monitor.pcx**

Localização: Centro Esquerda Direita

Nome do Nó:	Dispositivos de armazenamento	Número:	1,2,4
Contexto:	HARDWARE		
Background:	PADRÃO		

 Texto


Conteúdo	Os dispositivos de armazenamento guardam programas e seus trabalhos, que por razões de espaço não ficaram na memória principal. Os dispositivos de armazenamento mais comuns são: disco rígido - Winchester - e disquetes.		

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1	_____	5
2	_____	6
3	_____	7
4	_____	8

Obs: JANELA

 Som

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:

 Imagem/Vídeo

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo: teclado.pcx; mouse.pcx e monitor.pcx

Localização: Centro Esquerda Direita

Nome do Nó: MAINFRAMES	Número: 1,3
Contexto: HARDWARE	
Background: PADRÃO	

 **Texto**

Conteúdo	
	Mainframes são computadores de grande porte e são usados para fins comerciais e científicos onde existe a necessidade de processamento da informação.
	Folhas de pagamento de grandes empresas, operações financeiras de bancos e controle do sistema nacional de saúde, por exemplo, são processados por mainframes.
	Mainframe obedecem a uma mesma arquitetura básica.
	A evolução dos mainframes deu origem aos supercomputadores.

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1	Arquitetura básica	1,3,1	5
2	Supercomputadores	1,2	6
3			7
4			8

Obs: /

 **Som**

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:

 **Imagem/Vídeo**

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo: ARQ_PC.PCX

Localização: Centro Esquerda Direita

Nome do Nó: MICROCOMPUTADORES	Número: 1,4
Contexto: HARDWARE	
Background: PADRÃO	

 Texto

Conteúdo	Os microcomputadores surgiram na década de 80, graças aos avanços da tecnologia de microeletrônica. Os microcomputadores se caracterizam por seu tamanho reduzido e por possuírem grande capacidade de armazenar e processar informações. Por estas características os micros se popularizaram tornando-se computadores pessoais. Os microcomputadores também se caracterizam por possuírem uma mesma arquitetura básica.
----------	--

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1 computadores pessoais	1,4,1	5
2 arquitetura básica	1,2	6
3		7
4		8

Obs: /	
--------	--

 Som

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição	
-----------	--

 Imagem/Vídeo

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo:

Localização: Centro Esquerda Direita



Nome do Nó: PC's

Número: 1,4,1

Contexto: HARDWARE



Texto

Conteúdo

Histórico dos PC's

Ligações (NomeBotão - NúmeroNóDestino)

A -

E -

B -

F -

C -

G -

D -

H -

Obs:

JANELA



Som

Descrição:



Imagem/Vídeo

Descrição:

FOTO DE UM PC

Nome do Nó: SOFTWARE	Número: 2
Contexto: SOFTWARE	
Background: PADRÃO	

 Texto


Conteúdo	Software são programas executados pelo computador. Programas aplicativos e sistemas operacionais são tipos de software que nós utilizamos para resolver nossos problemas ou realizar tarefas.
	Programas são escritos usando linguagens de programação.

Alinhamento: Esquerda Centro Direita Justificado

Ligações (NomeBotão - NúmeroNóDestino)

1 Programas aplicativos	2,02	5
2 Sistemas operacionais	2,1	6
3 Linguagens de programação	2,01	7
4 Computador	1,0	8

Obs:

 Som

Tipo: Voz Humana Música

Nome do Arquivo:

Descrição:

 Imagem/Vídeo

Extensão: PCX BMP GIF AVI Outras:

Nome do Arquivo:

Localização: Centro Esquerda Direita