

Cristina Duarte Murta

Depto. Ciencia de Comp.  
ICEX - UFMG  
Convênio n.º Doação  
Data: 27/02/91  
Ass. Gen. Reneo

512.545(0)  
M.98.70  
1200

UMA INTERFACE PARA USUARIOS NOVATOS

DO SISTEMA PATPLUS

U. F. M. G. - BIBLIOTECA UNIVERSITÁRIA

OK  
2002  
SA



533239102

NÃO DANIFIQUE ESTA ETIQUETA

03

Dissertação apresentada ao Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Belo Horizonte

1990



FOLHA DE APROVAÇÃO

UMA INTERFACE PARA USUÁRIOS NOVATOS DO SISTEMA PATPLUS

CRISTINA DUARTE MURTA

Dissertação defendida e aprovada pela banca examinadora  
constituída pelos Senhores:

Prof. NÍVIO ZIVIANI - Orientador

Prof. IVAN MOURA CAMPOS

Prof. OSVALDO SÉRGIO FARHAT DE CARVALHO

Belo Horizonte, 25 de outubro de 1990.

A mim.

## AGRADECIMENTOS

A Gustavo, pelo incentivo e apoio irrestritos e por proporcionar em casa um ambiente propício ao estudo, agradável e tranquilo.

A Suzana, pelas conversas que provocaram inúmeras reflexões sobre o fundamento e relacionamento humanos e a vida.

Ao professor Nívio Ziviani, pela oportunidade deste trabalho e pela convivência sempre estimulante e modelar.

A meus pais, por tudo que me ensinaram.

Aos professores Ivan M. Campos e Regina H. E. Cabral, pelas preciosas sugestões.

Aos demais professores e colegas do DCC, por contribuírem direta ou indiretamente para minha formação e para este trabalho.

Ao CNPq, pelo apoio financeiro.

## Lições Mais Duras

Só "aprenderam" lições daqueles  
que admiravam vocês  
e eram ternos com vocês  
e que se punham de lado  
para dar a vez a vocês?

Não aprenderam as grandes lições  
daqueles que não aceitam vocês  
e dão-se os braços contra vocês  
e que tratam vocês com pouco caso  
ou que disputam a vez com vocês?

Walt Whitman

Se pudesse, viveria novamente minha vida.  
Na próxima, trataria de cometer mais erros.  
Não tentaria ser tão perfeito, relaxaria mais.

...

Correria mais riscos, faria mais viagens, contemplaria mais  
entardeceres, subiria mais montanhas, nadaria mais rios.  
Iria a mais lugares onde nunca houvesse ido, teria mais  
problemas reais e menos imaginários.

Jorge Luis Borges

Valeu a pena?  
Tudo vale a pena, se a alma não é pequena.

Fernando Pessoa

## RESUMO

E apresentada uma interface para usuários novatos do sistema PATPLUS, um sistema de recuperação de informação em texto completo, para microcomputadores. O projeto da interface foi fundamentado em um estudo da literatura existente sobre o assunto, que também é mostrado. Este estudo engloba os aspectos de classificação de usuários, estilos de interação e outras questões importantes no projeto de uma interface homem-computador.

## ABSTRACT

This thesis presents a human-computer interface for novice users of PATPLUS, a full-text information retrieval system for microcomputers. The design of the interface was based on a study of the literature on human-computer interfaces and interactive systems development. The main concepts and some aspects of dialogue design, interaction techniques and user types are described.

## SUMARIO

1	INTRODUÇÃO.....	1
1.1	Motivação.....	1
1.2	Trabalho Relacionado.....	3
1.3	Contribuições e Organização da Dissertação.....	5
2	OS USUARIOS E OS ESTILOS DE INTERAÇÃO.....	7
2.1	Classificação dos Usuários.....	7
2.2	Classificação dos Estilos de Interação.....	12
2.2.1	Diálogo Sequencial.....	13
2.2.2	Diálogo Assíncrono.....	20
3	ASPECTOS IMPORTANTES NO DESENVOLVIMENTO DE INTERFACES ..	24
3.1	Introdução.....	24
3.2	Independência do Diálogo.....	28
3.3	Assistência ao Usuário.....	30
3.4	Apresentação da Informação na Tela.....	35
3.5	Avaliação de Interfaces.....	42
4	ESPECIFICAÇÃO DE UMA INTERFACE PARA O SISTEMA PATPLUS...	48
4.1	O Sistema PATPLUS.....	48
4.1.1	Descrição do Sistema.....	48
4.1.2	A Linguagem de Consulta.....	50
4.2	Objetivos do Projeto.....	54
4.3	Descrição Estrutural da Interface.....	55



4.4	Representação da Interface.....	58
4.4.1	Interface do Módulo Principal.....	63
4.4.2	Interface do Módulo de Conversão e Construção.	65
4.4.3	Interface do Módulo de Recuperação.....	68
5	PROJETO DA INTERFACE PARA O SISTEMA PATPLUS.....	75
5.1	Arquitetura de Um Sistema Interativo.....	75
5.2	Modularização Interface - Sistema PATPLUS.....	77
5.2.1	Componente do Diálogo.....	78
5.2.2	Componentes de Controle Global.....	80
6	IMPLEMENTAÇÃO DA INTERFACE PARA O SISTEMA PATPLUS.....	84
7	EXEMPLO DE UTILIZAÇÃO DO SISTEMA PATPLUS.....	92
8	CONCLUSÃO.....	108
	REFERENCIAS BIBLIOGRAFICAS.....	110

## INTRODUÇÃO

### 1.1 Motivação

Sistemas de computação podem ser considerados parte de uma tecnologia nova, ainda no estágio inicial de disseminação e refinamento. Como máquinas fotográficas e automóveis, os computadores deverão, dentro de pouco tempo, estar disponíveis para todas as pessoas e não somente para aquelas que se dedicam a estudá-los. E neste contexto, no qual as possibilidades desta máquina são limitadas não por sua potência de cálculo mas por sua potência de comunicação com usuários humanos, que aparece com importância a questão do desenvolvimento de interfaces homem-computador.

A tentativa de melhorar a qualidade da utilização pelo homem, de seus produtos, está no domínio da disciplina de *fatores humanos*, também conhecida como *ergonomia*. No estudo da interação homem-computador os fatores humanos têm recebido importância crescente. Duas razões explicam este fato. A primeira é a expansão da população usuária, consequência da redução no custo de produção da máquina e do aumento de sua velocidade e confiabilidade, ambos em decorrência dos avanços tecnológicos, possibilitando o uso de computadores para numerosas finalidades. Atualmente, a população usuária é composta por um conjunto amplo e variado de tipos de pessoas.

A segunda razão é o aumento da dependência das organizações em relação aos sistemas interativos, tais como sistemas de reserva de hotéis e passagens aéreas, aplicações bancárias, serviços de recuperação de informação e sistemas de suporte à decisão. Em tais ambientes erros frequentes, tempos de resposta longos, baixo desempenho e operações confusas são inaceitáveis.

Com o advento das telecomunicações controladas via

computador, a disponibilidade crescente de documentos na forma eletrônica e o acesso a grandes bancos de dados, estima-se que, em futuro relativamente próximo, a maioria dos membros da comunidade terá uma rotina baseada em interações com sistemas de computador. Neste contexto, o projeto da interface de um sistema é crucial para sua eficiência e aceitabilidade e, portanto, para o seu potencial comercial.

A interface homem-computador é frequentemente reconhecida como sendo o fator mais importante na determinação do sucesso ou fracasso de um sistema. No desenvolvimento do projeto da interface é fundamental considerar as capacidades, habilidades e limitações do usuário. A interface tem também um custo significativo em relação ao custo total do sistema. Pesquisas mostram que o tamanho de código requerido para suportar a interface com o usuário está entre 30% e 60% do código total da aplicação (Baecker & Buxton, 1987; Britts, 1987).

O projeto de interfaces computador-usuário não está apenas no domínio da ciência da computação. É um processo multidisciplinar, que envolve disciplinas de psicologia cognitiva, educação, linguística, ergonomia, comunicação visual e projeto gráfico. O desafio básico é projetar interfaces que façam uma comunicação eficiente e precisa entre o usuário e o computador. A interface deve ser transparente para que o usuário possa dedicar toda a sua atenção para a execução de sua tarefa.

O objetivo desta dissertação é apresentar uma interface para o sistema PATPLUS, um sistema de recuperação de informação em texto completo desenvolvido no Departamento de Ciência da Computação da UFMG (Ziviani, 1990). Esta interface apresenta características especialmente projetadas para usuários novatos no uso desse sistema.

O sistema Patplus foi escolhido porque, devido às suas características, possui aplicabilidade ampla e geral, com grande número de usuários potenciais, o que torna a interface uma parte importante dentro do sistema como um todo.

A importância específica do usuário novato para o sistema Patplus reside em dois fatores: o quantitativo e o qualitativo. O aspecto quantitativo se apresenta no grande número de usuários

potenciais do sistema que serão, em primeiro plano, usuários novatos no uso do sistema. O aspecto qualitativo aparece quando se percebe que o sucesso do sistema reside na avaliação do seu uso feita pelo usuário novato. O usuário deve sentir que o resultado obtido com o uso do sistema compensa satisfatoriamente o esforço despendido no seu uso. E é através da interface que o usuário "vê" o sistema. Deve-se portanto, a estes fatores, a motivação desta dissertação.

## 1.2 Trabalho Relacionado

Os fundamentos para o desenvolvimento de interfaces homem-computador eficientes surgiram durante a última década. Entre os responsáveis por este trabalho inicial estavam alguns estudiosos de computação gráfica (Foley & Wallace, 1974; Newman & Sproull, 1979). Desde então, grande parte do trabalho produzido foi dedicado ao estudo de princípios e linhas gerais de projeto de interfaces (Baecker & Buxton, 1987; Gould & Lewis, 1985; Norman, 1983; Rubinstein & Hersh, 1984) e este trabalho não foi, em geral, comprovado experimentalmente. Das centenas de linhas gerais compiladas da literatura somente uma pequena parte tem base empírica, encontrando-se muitas contradições e inconsistências. Não obstante, princípios e linhas gerais de projeto são importantes e existe a necessidade de metodologias e ferramentas que facilitem a inclusão destes princípios no projeto da interface.

O interesse em engenharia humana para sistemas de computação tem crescido a ponto de números inteiros de periódicos serem dedicados à pesquisa nesta área (ACM *Computing Surveys*, 1981; ACM *Computing Surveys*, 1989; *Communications of the ACM*, 1983; IBM *Systems Journal*, 1981; IEEE *Computer*, 1982; IEEE *Software*, 1989). Simpósios são dedicados ao estudo da interface homem-computador e seu projeto (por exemplo, ACM SIGGRAPH, 1986; ACM SIGGRAPH, 1988). Uma sequência de encontros neste campo levou às conferências CHI (*Computer-Human Interaction*) e outras similares (ACM CHI 1983, 1985, 1986, 1987, 1988; HCI Hawaii 1984, 1987; INTERACT 1984, 1987). Periódicos (*International Journal of*

*Man-Machine Studies*; *SIGCHI Bulletin*) e livros (Baecker & Buxton, 1987; Monk, 1984; Shneiderman, 1987; Vassiliou, 1984) especializados na área foram lançados.

Um artigo particularmente importante é o de Hartson & Hix (1989), que aborda conceitos essenciais no gerenciamento de interfaces homem-computador, estabelecendo uma estrutura para classificação e comparação de abordagens para o problema do gerenciamento destas interfaces. Estes conceitos são apresentados no capítulo 3. Neste artigo são também apresentados alguns modelos estruturais de interfaces homem-computador, alguns métodos para representação de interfaces e várias ferramentas interativas que auxiliam o desenvolvimento de interfaces. Outra contribuição relevante é a descrição de um modelo de interação homem-computador e de técnicas para representação do diálogo, apresentada em Hartson, Hix & Kraly (1990).

Baecker & Buxton (1987) reuniram dezenas de artigos publicados em um livro de setecentas páginas onde aspectos da interação homem-computador são abordados. Monk (1984) focaliza aspectos de projeto de interfaces, de estudo sobre usuários e de avaliação da interface, mostrando como e quando colher dados experimentais e como fazer uma avaliação estatística desses dados. Shneiderman (1987) aborda o tema projeto de interfaces de forma abrangente e prática, apresentando estudos sobre projeto, implementação, gerenciamento, manutenção, treinamento e aperfeiçoamento da interface com o usuário de sistemas interativos, incluindo recomendações práticas e evidências experimentais. Vassiliou (1984) reúne treze trabalhos apresentados no '1982 New York University Symposium on User Interfaces'. São tópicos variados dentro do tema fatores humanos e sistemas de computador interativos.

Dentre as interfaces produzidas pela indústria, as mais revolucionárias foram aquelas que originaram o computador pessoal Apple Macintosh (Williams, 1984). Seguindo o sucesso dos projetos Smalltalk (Tesler, 1981) e Star (Smith, Kimball, Irby & Verplanck, 1982) da Xerox PARC e Apple LISA (Williams, 1983), o Macintosh introduziu novos conceitos (diálogo assíncrono, sistema multitarefa) e técnicas (janelas múltiplas, manipulação direta)

no campo da interação homem-máquina. O impacto destas inovações tem sido enorme mudando de maneira definitiva o conceito e a aparência das interfaces. Desde então esses projetos são considerados modelos e são copiados para outros computadores, estações de trabalho e mesmo *mainframes*. Alguns conceitos e técnicas introduzidos por esses sistemas são apresentados no capítulo 2.

Um fato importante e recente é o surgimento de ferramentas interativas que auxiliam o projetista na criação e no gerenciamento de muitos aspectos da interface. Estas ferramentas são denominadas sistemas de gerenciamento de interfaces com o usuário e são conhecidas como UIMS (*User Interface Management Systems*) (Foley, 1987; Hartson & Hix, 1989; Pfaff, 1985). Os UIMS fornecem ao projetista e ao programador um ambiente para produzir interfaces sem precisar de depender somente das bibliotecas de subrotinas e das linguagens de programação. Os UIMS têm como objetivo automatizar a produção de software para interfaces com o usuário, e têm como saída (produto) código de programa que pode ser executado para produzir a interface ou descrições declarativas que podem ser interpretadas também para produzir a interface. As atividades de gerenciamento da interface com o usuário (gerenciamento do diálogo) consistem dos aspectos de desenvolvimento da interface homem-computador incluindo representação, projeto, implementação, prototipação, execução, avaliação e manutenção. As ferramentas de desenvolvimento de interfaces frequentemente incluem editores de diagramação de estados, editores gráficos, editores de texto, sistemas de gerenciamento de bancos de dados e prototipadores.

### **1.3 Contribuições e Organização da Dissertação**

A principal contribuição desta dissertação é o estudo de interfaces para usuários novatos, através da especificação, do projeto e da implementação de uma interface para o sistema Fatplus. Esta interface é dedicada aos usuários novatos no uso deste sistema e é baseada em janelas e menus. No entanto, pode ser usada pelos usuários experientes porque é mais agradável e

mais robusta que a anterior. Quanto à rapidez na execução das tarefas podemos dizer que a interface baseada em seleção de menus pode ser competitiva com a interface baseada em linguagem de comandos.

Outra contribuição é a elaboração de um esquema geral para especificação de interfaces para usuários novatos de sistemas de recuperação de informação em texto completo, descrito no capítulo 4. O Patplus é um exemplo típico destes sistemas. A especificação da interface pode ser generalizada para outros sistemas deste grupo.

Também são descritas, no capítulo 3, algumas linhas gerais úteis na especificação, no projeto e na implementação de interfaces de caráter genérico, que podem ser consideradas por projetistas de software que necessitem desenvolver alguma interface.

Outra contribuição, que pode ser encarada mais como uma consequência do trabalho desenvolvido, é a reunião em um único texto de aspectos teóricos importantes no projeto de interfaces com ênfase para as interfaces adequadas aos usuários novatos e inexperientes no uso de um sistema, além de vasta bibliografia sobre o tema. Alguns aspectos deste estudo são apresentados nos capítulos 2 e 3.

Esta dissertação está organizada em sete capítulos, dos quais este é o primeiro. O capítulo 2 apresenta uma classificação de usuários e uma classificação para os estilos de interação. O capítulo 3 trata de aspectos importantes no desenvolvimento de interfaces homem-computador que são independência do diálogo, assistência ao usuário, apresentação da informação na tela e avaliação. O capítulo 4 descreve rapidamente o sistema Patplus, apresenta os objetivos deste projeto, descreve a especificação da interface e mostra a representação de seus módulos. O capítulo 5 mostra o projeto da interface e o capítulo 6 apresenta detalhes de sua implementação. O capítulo 7 apresenta exemplos de utilização do sistema com apresentação das telas. Finalmente a conclusão do trabalho é apresentada no capítulo 8, seguida das referências bibliográficas.

## CAPITULO 2

### OS USUARIOS E OS ESTILOS DE INTERAÇÃO

#### 2.1 Classificação dos Usuários

O estudo dos usuários de sistemas interativos é fundamental no projeto de interfaces. O usuário é o componente crítico que determina se o sistema homem-computador como um todo funciona ou não. A importância do estudo dos usuários é óbvia, especialmente quando se considera a disseminação dos sistemas interativos para novas comunidades de usuários, o que torna desnecessário qualquer argumento a seu favor.

O estudo dos usuários pode ser dividido em duas fases. A primeira é a determinação do modelo mental dos usuários (Eberts & Eberts, 1989; Norman, 1987) em relação ao sistema. Modelo mental é o modelo que o usuário forma sobre como um sistema de computação ou um programa de aplicação trabalha. O modelo mental é formado na mente do usuário enquanto ele utiliza o sistema ou mesmo antes de sua utilização consistindo, neste caso, na idéia que o usuário faz do sistema. O modelo mental possibilita o usuário entender e interagir com o sistema.

Contrapondo-se ao modelo mental do usuário está o modelo conceitual do sistema, que é o conjunto de conceitos que forma a base do comportamento do sistema juntamente com as funções que o sistema executa. O modelo conceitual é a descrição do sistema de computação em termos de engenharia e, portanto, ele é preciso, consistente e completo. Em um projeto orientado para fatores humanos, o modelo conceitual do sistema deve ser o mais próximo possível do modelo mental do usuário porque, dessa forma, o projetista pode utilizar o conhecimento que o usuário "já possui" em relação ao sistema, tornando mínimo o aprendizado ou a mudança de conceitos necessária para seu uso. O modelo mental reflete a expectativa do usuário em relação ao sistema. Portanto, este modelo deve ser ponto de partida na elaboração do modelo



conceitual do sistema.

Embora, segundo Norman (1987), os modelos mentais sejam incompletos, instáveis e sem fronteiras fixas, estes modelos vão tornando-se mais completos e precisos à medida que o usuário interage com o sistema. Esta mudança pode repercutir na interface através da técnica de projeto interativo e desenvolvimento de protótipos, que será apresentada no capítulo 3. O projeto final deve exibir consistência entre estes modelos.

A segunda fase do estudo dos usuários é a classificação da população usuária em termos de sua experiência acumulada na interação com sistemas de computação. Esta análise auxilia o projetista na determinação dos estilos de interação que devem compor a interface. Existem diversas propostas para classificação dos usuários de sistemas de computação e, embora haja divergências, muitos autores concordam que o modelo de classificação deve ser dinâmico uma vez que os usuários mudam de tipo no tempo. A asserção de que usuários especialistas e profissionais enfatizam usabilidade enquanto usuários novatos e casuais enfatizam facilidade de aprendizado é também amplamente aceita.

Uma análise das propostas de Carey (1982), Greenberg & Witten (1988), Schneider (1984), Shneiderman (1987) e Vassiliou & Jarke (1984) revela que as quatro primeiras podem ser derivadas da proposição de Vassiliou & Jarke, que é baseada num sistema composto de quatro variáveis binárias: familiaridade com conceitos de programação, frequência de uso do sistema, conhecimento da aplicação e faixa de operações.

*Familiaridade com conceitos de programação* avalia as habilidades algorítmica e lógica dos usuários, bem como o grau de conhecimento dos conceitos de objetos (diretórios, arquivos, linhas, caracteres, números binários) e ações (criar, inserir, salvar, apagar, imprimir) no mundo da computação. 'Muita' familiaridade com conceitos de programação é assinalada para usuários que não têm medo de computadores e que têm experiência em raciocínio algorítmico e lógico na solução de problemas.

A variável *frequência de uso do sistema* está relacionada com a quantidade de treinamento. Esta dimensão é importante

especialmente nos casos em que a frequência de uso é tão baixa que a cada utilização do sistema é necessário o reaprendizado ou a rotatividade de usuários é alta, isto é, a proporção de usuários novatos no sistema é significativa a qualquer tempo.

A combinação das duas variáveis discutidas indica a capacidade do usuário de interagir tecnicamente com o sistema e representa a dimensão sintática da interação. Esta dimensão leva em consideração apenas a relação usuário-computador, o usuário tendo o computador como perspectiva de ferramenta. Em última análise ela mede a capacidade do usuário de utilização da ferramenta computador. A Tabela 2.1 mostra o relacionamento entre estas duas variáveis básicas e a classificação dos usuários no nível sintático da interação. Três tipos de usuários são derivados: novato, habilitado e profissional. O termo 'novato' pode ser assinalado não somente para novos usuários, mas também para usuários infrequentes com pouco conhecimento de programação.

**TABELA 2.1**

**Tipos de Usuários**

**Conhecimento Sintático - Capacidade de Interação**

Frequência de uso do sistema	Familiaridade com conceitos de programação	
	pouca	muita
baixa	novato	habilitado
alta	habilitado	profissional

A terceira variável diz respeito ao *conhecimento da aplicação* e mede a precisão do modelo mental dos usuários em relação ao modelo conceitual do programa de aplicação em termos de sua estrutura e conteúdo. O conhecimento da aplicação varia de uma 'idéia geral' a um 'conhecimento detalhado'.

A última variável é a *faixa de operações* e descreve a porcentagem de operações de um programa de aplicação que o usuário utiliza. Juntas, as variáveis conhecimento da aplicação e faixa de operações dão uma visão da capacidade de uso da aplicação e compõem a dimensão semântica da interação. Esta dimensão mede a capacidade do usuário de execução de uma tarefa utilizando um determinado programa de computador (aplicação). A Tabela 2.2 mostra a classificação dos usuários no nível semântico da interação. Quatro tipos de usuários são derivados dessa classificação: (1) usuário novato em relação à capacidade de uso da aplicação é aquele que utiliza o sistema através de um conjunto limitado de funções, (2) usuário administrativo é o que executa um conjunto limitado de tarefas mas que tem um conhecimento detalhado de sua aplicação, (3) usuário gerencial, que tem pouco interesse em detalhes mas que requer ampla funcionalidade, e (4) especialista na aplicação, que tem um conhecimento excelente da aplicação e executa uma faixa vasta de tarefas.

**TABELA 2.2**

**Tipos de Usuários**

**Conhecimento Semântico - Capacidade de Uso da Aplicação**

Faixa de operações	Conhecimento da Aplicação	
	geral	detalhado
estreita	novato	administrativo
ampla	gerencial	especialista na aplicação

Cada tipo de usuário exige interfaces com características bastante diferentes. O usuário novato enfatiza:

- (a) vocabulário restrito a um pequeno número de termos significativos e familiares;
- (b) retorno informativo sobre o cumprimento de cada tarefa;
- (c) pequeno número de opções de tarefas;
- (d) proteção contra qualquer erro possível;
- (e) mensagens de erro específicas e construtivas;
- (f) manuais projetados com seu vocabulário e com exemplos explicativos.

O usuário administrativo prefere:

- (a) muitas opções de pequenas tarefas;
- (b) pouca informação por vez na tela;
- (c) retorno informativo para tarefas longas ou complexas;
- (d) proteção contra erros sérios;
- (e) mensagens de erro construtivas;
- (f) sistema de ajuda em linha e manuais de usuário detalhados.

O usuário gerencial requer:

- (a) comandos distintos e significativos;
- (b) sequências consistentes de ações;
- (c) acesso fácil e rápido às tarefas;
- (d) sistemas auto-explicativos;
- (e) retorno informativo relacionado à tarefa: para ações frequentes, respostas simples e para ações infrequentes ou complexas, respostas mais substanciais;
- (f) proteção contra erros sérios.

Usuários especialistas escolhem sistemas com:

- (a) tempos de resposta rápidos;
- (b) retorno informativo breve;
- (c) comandos que exijam pouca digitação ou seleção;
- (d) possibilidade de criar macros e abreviações dos comandos mais utilizados para acelerar a execução das tarefas;
- (e) manuais de referência completos e impressos.

A variedade destas características indica que uma única interface não acomoda as diferenças individuais dos usuários. Portanto, projetar uma interface para todos os tipos de usuário pode ser uma tarefa impossível.

O processo de interação usuário-computador pode ser otimizado quando as características relevantes da população usuária do sistema são conhecidas. O projetista deve ter em mente que a comunidade usuária é uma fonte valiosa de informações sobre o sistema e deve utilizar esta fonte solicitando comentários e sugestões sobre os aspectos de projeto e operação da interface. O valor de um esquema de classificação de usuários está também no fato que a metodologia de avaliação da interface pode ser derivada desse esquema.

## 2.2 Classificação dos Estilos de Interação

No projeto da interface o projetista deve escolher um ou mais estilos de interação para compor a interface. Esta escolha é função das tarefas que serão implementadas na aplicação, do custo da interface, do hardware disponível e dos usuários potenciais do sistema. Para isto é necessário identificar os tipos básicos de diálogo homem-computador. Dentre as diversas propostas encontradas na literatura, duas são particularmente interessantes. A primeira é a de Hartson (1989) e a segunda é a de Jarke & Vassiliou (1985).

Hartson descreve duas metáforas através das quais os estilos de diálogo podem ser descritos: o mundo da conversação e o mundo da ação. O mundo da conversação, no qual o usuário descreve o que fazer através de uma linguagem de comandos, é relacionado ao diálogo sequencial, de contexto específico e previsível. O mundo da ação, no qual o usuário mostra o que fazer através da manipulação de representações visuais de objetos, é relacionado com o diálogo assíncrono, de caráter imprevisível porque suporta sistemas multitarefa. Exemplos de diálogo sequencial são os estilos de interação tradicionais e exemplos de diálogo assíncrono são os sistemas de manipulação direta.

Jarke & Vassiliou dividem os estilos de interação em dois grupos. O primeiro, denominado 'primeiras gerações' de estilos de interação, é caracterizado por um ambiente interativo restrito e estático onde o usuário tem como dispositivos de entrada e saída apenas o teclado e o terminal. Neste grupo estão todos os estilos

tradicionais de interação. O segundo grupo, denominado 'novas gerações' de estilos de interação, é caracterizado por uma tentativa de incrementar a utilização dos sentidos humanos com o objetivo de aumentar a produtividade, o interesse e o prazer e diminuir o "stress" durante a interação. Os autores incluem os sistemas de manipulação direta e as interfaces inteligentes neste grupo.

Jarke & Vassiliou descrevem e delimitam dois grupos de estilos de interação, mostrando inclusive seu desenvolvimento histórico. Hartson aponta a característica principal de cada grupo: diálogo sequencial e diálogo assíncrono. A partir desses dados é possível elaborar uma nova proposta que será descrita nos itens subsequentes. Esta proposta classifica os estilos de interação em dois grandes grupos: os estilos de interação baseados em diálogo sequencial e os estilos de interação baseados em diálogo assíncrono.

### **2.2.1 Diálogo Sequencial**

Diálogo sequencial é uma forma de diálogo que encaminha-se de maneira previsível de um parceiro do diálogo para o outro. O diálogo sequencial permite que usuários e projetistas visualizem um comportamento específico, lógico e sequencial. O sequenciamento é controlado pelo sistema, que apresenta uma tarefa por vez na tela. O tipo da entrada do usuário é esperado e o sistema usa seu valor para determinar a próxima sequência. As sequências de execução das transações são determinadas a priori pelo projetista. O diálogo sequencial tem um forte aspecto linguístico e as sequências válidas de tokens são usualmente descritas por gramáticas (por exemplo, BNF) e máquinas determinísticas de estados finitos.

A forma sequencial de diálogo é a característica básica das primeiras gerações de interfaces homem-computador e os estilos de interação que compõem este grupo desenvolveram-se a partir de duas diferentes linhas de pesquisa, conforme mostra a Figura 2.1. A primeira originou-se das concepções teóricas de linguagens de

programação e linguagens matematicamente orientadas e a segunda surgiu da necessidade de interfaces simples com o usuário final. O desenvolvimento dos estilos de interação pode ser representado por um gráfico bidimensional cujos eixos representam as capacidades funcionais de um sistema e sua usabilidade. Funcionalidade de um sistema refere-se às tarefas que um usuário pode executar com o sistema e usabilidade refere-se à facilidade na execução das tarefas. Esse gráfico é mostrado na Figura 2.2.

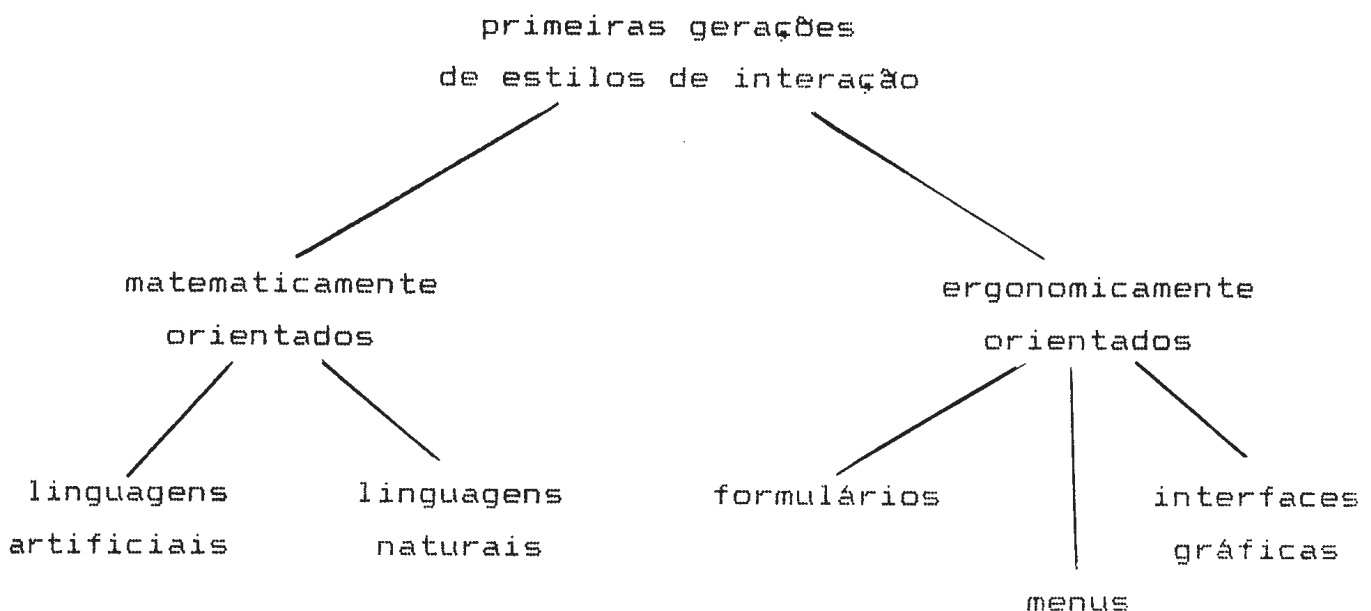


Fig. 2.1 Classificação dos Estilos de Interação baseados em Diálogo Sequencial

O grupo de estilos de interação originário das disciplinas de linguagens de programação e teoria de linguagens desenvolveu inicialmente as linguagens artificiais que compreendem as linguagens de comandos, linguagens de programação e linguagens de consulta a bancos de dados movendo-se então para o campo das linguagens naturais. A tendência é buscar mais usabilidade preservando a funcionalidade.

O segundo grupo iniciou a partir da análise ergonômica da

interação de usuários novatos com sistemas de computador. Deste estudo surgiram os estilos de preenchimento de formulários, menus e interfaces gráficas. Este desenvolvimento representa uma tendência para maior funcionalidade permanecendo a orientação para usuários novatos. A evolução dos estilos de interação ocorreu em função dos progressos nas áreas de tecnologia de hardware, computação gráfica, inteligência artificial e psicologia aplicada à ciência da computação. Os estilos de interação baseados na forma sequencial de diálogo são apresentados a seguir segundo os itens Descrição, Aplicação e Questões de Projeto.

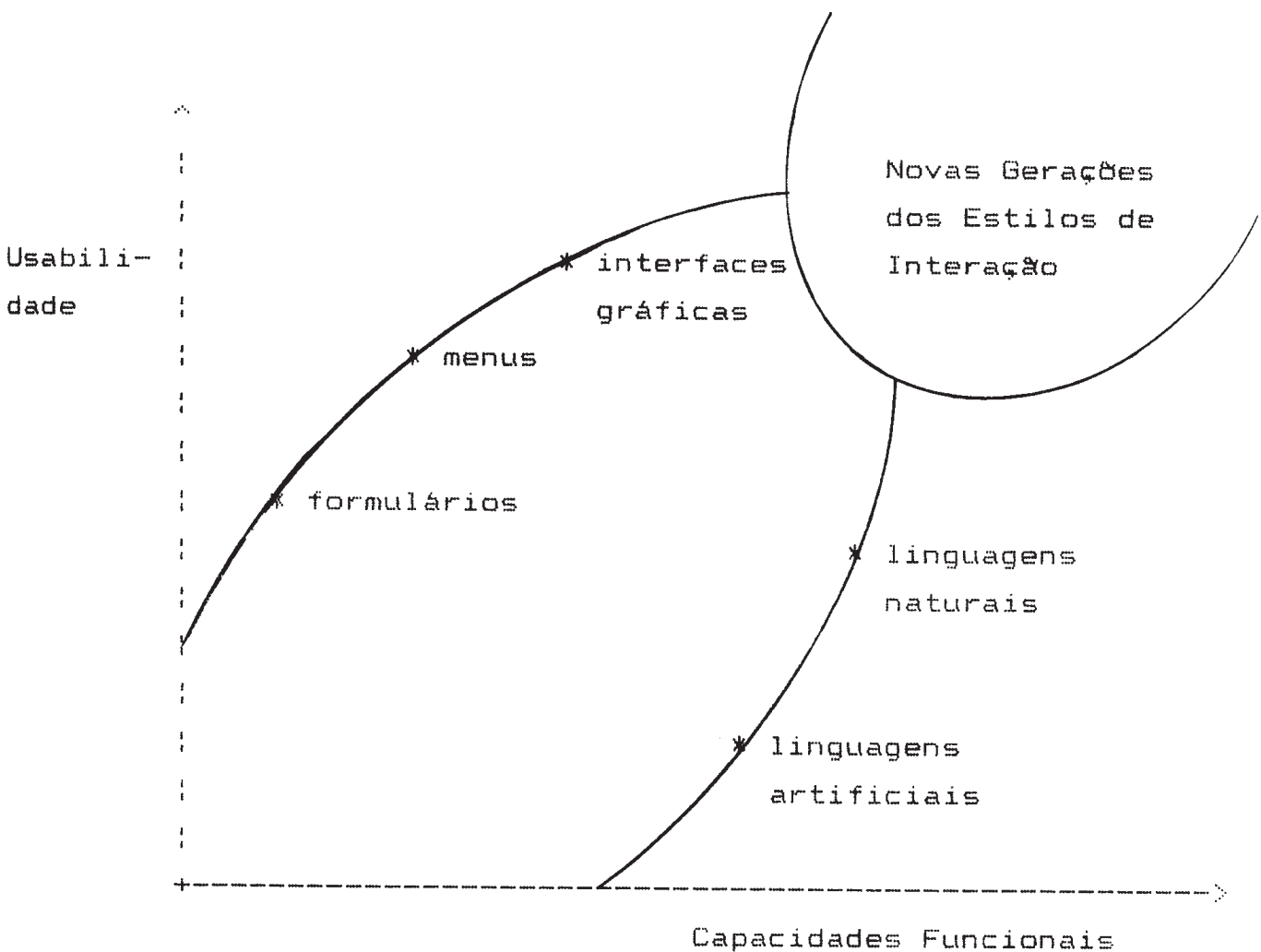


Figura 2.2 Evolução dos Estilos de Interação



## Linguagens Artificiais

Descrição: Os usuários digitam instruções para o computador utilizando comandos de um conjunto definido formalmente pela linguagem de comandos da aplicação. É o estilo mais tradicional e teve origem com os sistemas operacionais. Os comandos são imediatos e de impacto sobre os dispositivos ou a informação dando ao usuário as sensações de potência, iniciativa, controle e rapidez. Os comandos são breves e sua existência é transitória. Linguagens de comandos podem consistir de comandos únicos ou ter uma sintaxe complexa e podem ter poucas ou centenas de operações.

Aplicação: Devido à necessidade de aprendizado da sintaxe, que pode ser complexa, é apropriada para usuários especialistas. As taxas de erros são tipicamente altas, o treinamento é necessário e a retenção da sintaxe dos comandos é baixa. A maioria dos erros é devida a comandos ilegais e estes erros são de fácil detecção e fácil correção. Neste estilo, assistência em linha pode ser difícil de fornecer devido à diversidade de possibilidades. Não exige hardware especial.

Questões de projeto: As questões mais estudadas são o projeto de uma terminologia ótima, a funcionalidade do conjunto de comandos e as estratégias para organização dos comandos.

O projeto da terminologia é importante para o aprendizado, a solução dos problemas durante a interação e a retenção ao longo do tempo. Os nomes dos comandos devem ser significativos e escolhidos em função do conjunto.

A funcionalidade de um sistema representa a sua potência. Funcionalidade insuficiente pode deixar o usuário frustrado porque uma função aparentemente óbvia não é suportada e o excesso de funcionalidade implica mais código para manter, maiores possibilidades de erro, possivelmente execução mais lenta, mais telas de ajuda, mensagens de erro e manuais mais longos. Isto significa para o usuário aprendizado mais lento, aumento das chances de erro, manuais mais longos e mensagens de erro menos específicas.

As estratégias para organização de comandos são escolhidas com o objetivo de ajudar no aprendizado, na solução de problemas e na retenção ao longo do tempo. O uso de macros e de menus de comandos deve também ser considerado.

## **Linguagens Naturais**

Descrição: Estilo no qual a linguagem de comandos é um subconjunto bem definido e significativo de alguma linguagem natural, como por exemplo o inglês. Embora cada termo da linguagem possa ser conhecido do usuário, este deve aprender qual é o subconjunto válido em relação à linguagem origem e qual a sintaxe permitida.

Aplicação: A interface em linguagem natural é adequada para programas com grande complexidade semântica, isto é, programas com grande número de tarefas executáveis e que necessitem enviar muitas mensagens. Este tipo de programa necessita de uma linguagem complexa, que ofereça diversas opções e que seja fácil de aprender. A linguagem natural é, então, adequada porque preenche todos estes requisitos. A linguagem natural não é adequada sob o ponto de vista do custo da interface (Rich, 1984) porque o projeto e a implementação são caros e o tempo de execução é maior. Para sistemas onde a precisão é um fator importante ela não deve ser empregada porque pode gerar declarações ambíguas.

Questões de Projeto: A principal questão é a dificuldade de implementação de um conjunto grande de palavras. As interfaces produzidas em linguagem natural contêm um vocabulário pequeno que não possibilita uma interação efetiva. Outra questão é o tempo dedicado à interação. O usuário deve digitar frases inteiras e não apenas comandos para comunicar sua consulta. Além disso, ele está sujeito a um diálogo de esclarecimento que verifica se as suas intenções foram reconhecidas. Este tempo é muito longo quando comparado com os demais estilos de interação, em especial o de manipulação direta, mesmo se houver reconhecimento de voz.

## **Formulário**

Descrição: É uma tela de requerimentos em que várias opções e valores são especificados e integrados à tela através do preenchimento de campos. Poucas instruções são necessárias, uma vez que a noção de preenchimento de formulários é familiar. É um estilo atrativo porque a visibilidade da informação dá aos usuários um sentimento de controle do diálogo. Este estilo de interação é a nova tendência para interfaces de bancos de dados.

Aplicação: São interfaces utilizadas quando a aplicação requer a entrada de grande volume de dados. Usuários devem ter desenvoltura no uso do teclado e devem conhecer o significado dos títulos dos campos e o conjunto de respostas possíveis para cada campo. Os usuários devem conhecer também os mecanismos de correção de erros.

Questões de Projeto: A qualidade das interfaces baseadas em formulários está ligada a dois fatores. O primeiro é o quanto a lógica do formulário retrata a lógica do sistema para o qual está estruturando a entrada de dados e o segundo é a clareza do projeto e a apresentação visual das telas. O projeto de formulários deve considerar aspectos como título significativo, instruções compreensíveis, agrupamento e sequenciamento dos campos, nomes familiares para os campos, abreviações e terminologia consistentes, facilidade de correção de erros e auxílio em linha.

## **Menus**

Descrição: São sistemas nos quais os usuários editam comandos selecionando-os em uma lista de alternativas mostradas na tela. Se a terminologia e o significado dos itens são inteligíveis e distintos, os usuários podem cumprir suas tarefas com pouco treinamento, memorização e digitação. A grande vantagem dos menus é a apresentação de uma estrutura clara para tomada de decisão através de um conjunto limitado de opções.

Aplicação: Menus são, geralmente, mais lentos que linguagens de comandos especialmente se são compostos de muitas opções ou têm

uma estrutura hierárquica. Podem ser auto-explicativos, não requerendo conhecimento prévio do sistema. O aprendizado e a utilização são fáceis e estimulam a exploração de opções. Este estilo de interação é apropriado para usuários com pouco ou nenhum treinamento ou usuários intermitentes no uso do sistema, mas pode ser atraente para especialistas se os mecanismos de seleção e apresentação na tela forem muito rápidos. McCracken & Akscyn (1984) relatam sua experiência com um sistema cuja interface baseada em seleção de menus é competitiva com a interface baseada em linguagem de comandos devido à rapidez de resposta do sistema. Além disto, usuários especialistas tendem a utilizar uma faixa muito mais ampla de funções do sistema do que os novatos e, portanto, apreciam o fato de o sistema de menus aliviar sua carga de memória.

Questões de Projeto: As questões que envolvem o projeto de menus são a sua organização semântica, a sequência de apresentação das opções, o tempo de resposta, a taxa de apresentação e o projeto da tela.

A organização semântica dos menus deve ser relevante, compreensível e distinta, de modo que o usuário sinta confiança ao fazer uma escolha. Os menus podem ser simples ou organizados em sequências lineares, estruturas em árvores e estruturas cíclicas e acíclicas.

A sequência de apresentação dos itens é uma decisão trivial se estes seguem uma ordem natural como os dias da semana ou os capítulos de um livro. Se esta ordem natural não existe, então o projetista deve escolher entre a sequência alfabética, o agrupamento de itens relacionados com utilização de separadores entre os grupos, a prioridade dos itens mais frequentemente usados ou a prioridade dos itens mais importantes. A péssima performance da sequência aleatória, comprovada experimentalmente (Shneiderman, 1987), confirma a importância de se considerar uma das alternativas apresentadas.

O projeto das telas de menus deve considerar aspectos de títulos das telas, nomes e seletores das opções e localização dos itens, das instruções e das mensagens de erro.

## **Interação Gráfica**

Descrição: O usuário define, modifica e trabalha com desenhos, diagramas, imagens e figuras bidimensionais e tridimensionais.

Aplicação: A interação gráfica pode ser aplicada quando os objetos de trabalho do usuário podem ser expressos de maneira natural por figuras, desenhos ou gráficos e as tarefas podem ser desempenhadas por ações tais como apertar botões, selecionar e localizar posições.

Questões de Projeto: De acordo com Baecker (1980), o projeto de um sistema gráfico deve ser baseado em um exame cuidadoso de seis fatores: tecnologia de exibição, tecnologia de entrada de dados, os parceiros do diálogo (o usuário e a máquina), o conteúdo e o contexto do diálogo.

### **2.2.2 Diálogo Assíncrono**

O diálogo de controle assíncrono viabiliza a implementação de sistemas multitarefa, caracterizados pela multiplicidade de opções (caminhos) disponíveis para o usuário a um dado instante durante o diálogo. O diálogo é assíncrono no sentido de que o sequenciamento de uma tarefa é independente das outras. A qualquer ponto durante a execução de uma tarefa o usuário pode desviar para outra tarefa e, posteriormente, retornar à primeira. O projetista não tem conhecimento a priori de que tipo de entrada o usuário escolherá. Este tipo de diálogo é usualmente descrito em termos de objetos e ações. Cada item léxico (*token*) em um diálogo assíncrono é conectado com um evento que resulta de uma ação do usuário e causa uma ação do sistema em resposta. Mecanismos baseados em eventos (Green, 1986) fornecem técnicas para controle e comunicação sobre as quais o diálogo assíncrono é construído.

A nova geração de estilos de interação é caracterizada pela ênfase tanto na funcionalidade quanto na usabilidade do sistema (Figura 2.2) e pela comunicação através de diálogos assíncronos. Esta geração é constituída pelas interfaces de manipulação direta e pelas interfaces inteligentes, conforme mostra a Figura 2.3. As

interfaces desse grupo podem ser aplicadas a praticamente qualquer tipo de sistema e são adequadas também para uma ampla faixa de usuários. O projeto de sistemas com interfaces assíncronas requer um estudo ainda mais cuidadoso do modelo mental dos usuários porque, neste caso, a comunicação homem-computador é mais rica e mais simbólica, podendo levar a desentendimentos mais graves. O surgimento destes estilos de interação é muito recente e a tecnologia que permite sua implementação está apenas começando a se delinear.

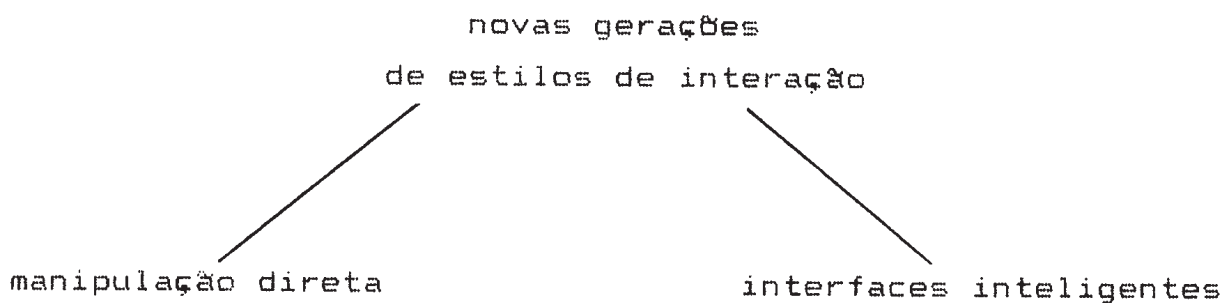


Fig. 2.3 Classificação dos Estilos de Interação baseados em Diálogo Assíncrono

Uma extensão do diálogo assíncrono é o diálogo concorrente, no qual mais de uma tarefa pode ser executada simultaneamente. Enquanto uma tarefa é executada outra pode ser inicializada, superpondo-se à primeira. Isto representa concorrência dos pontos de vista do usuário e do sistema. Para a interação concorrente são utilizadas janelas (*windows*) (Card, Pavel & Farrell, 1985), que são sistemas ou ambientes em que a tela é dividida em um número de áreas retangulares claramente delimitadas, com possível sobreposição. Cada janela pode ter um diálogo individual, tratar de uma função específica ou ser um 'terminal virtual', permitindo atividades múltiplas concorrentes em uma estação de trabalho. A

idéia básica é simples, funcionando como se o usuário tivesse um conjunto de terminais de diferentes tamanhos em sua estação de trabalho. Cada janela/terminal é uma visão particular de algum dado no computador. É uma técnica bastante difundida e utilizada em conjunto com os estilos de interação. Os estilos de interação baseados em diálogo assíncrono são descritos a seguir e algumas questões de projeto são apontadas.

### **Manipulação Direta**

Descrição: O usuário manipula uma representação gráfica dos dados utilizando uma linguagem de apertar botões e movimentar dispositivos de apontamento como, por exemplo, um "mouse". As idéias centrais identificadas por Shneiderman (1984) são visibilidade de objetos e ações, ações rápidas, incrementais, reversíveis e representadas por movimentos físicos. Os resultados são imediatamente visíveis e a linguagem de comandos é substituída pela manipulação direta do objeto de interesse. É um dos estilos mais atraentes, com interface agradável que oferece uma representação natural da tarefa e dos comandos que, por sua vez, são fáceis de aprender, usar e mantêm boa retenção no tempo.

Usando estes princípios é possível projetar sistemas com os seguintes atributos:

1. novatos podem aprender funções básicas rapidamente;
2. especialistas podem trabalhar de modo muito rápido executando uma grande variedade de tarefas;
3. usuários intermitentes retêm conceitos operacionais;
4. raramente são necessárias mensagens de erro;
5. usuários podem ver imediatamente se suas ações correspondem aos seus objetivos;
6. a ansiedade dos usuários é reduzida porque o sistema é compreensível e as ações são reversíveis.

Questões de Projeto: A questão básica é o projeto das representações gráficas visuais e espaciais incluindo estudo e validação do significado dos componentes da representação gráfica e alocação de espaço na tela. A escolha adequada de objetos e ações é uma tarefa difícil e deve ser validada através de testes.

## **Interface Inteligente**

Descrição: Interface inteligente é definida por Chignell, Hancock e Loewenthal (1989) como uma entidade inteligente que faz a mediação entre dois ou mais agentes em interação que possuem um entendimento recíproco incompleto do conhecimento e da forma de comunicação. Monk (1984) sugere que as interfaces inteligentes sejam vistas como um 'assistente sábio' cuja tarefa é ajudar e ensinar ao usuário a interagir de forma competente e eficiente com sistemas convencionais. Para isto, a interface deve ser sensível ao contexto da interação homem-computador e incorporar os seguintes tipos de conhecimento: conhecimento sobre o problema e sobre o domínio da aplicação, conhecimento sobre o sistema, o programa de aplicação e a própria interface, conhecimento sobre as técnicas de interação e conhecimento sobre o usuário. O conhecimento sobre o usuário pode ser obtido através de perguntas dirigidas a este no início da sessão ou por inferência direta do seu comportamento.

Questões de Projeto: Elkerton & Williges (1989) apontam a necessidade de um modelo de sistema para representar todos estes conhecimentos necessários e de um gerenciador de diálogos para coordenar os diálogos pertencentes a cada campo de conhecimento. Um projeto efetivo destes componentes requer uma pesquisa sistemática teórica e empírica sobre a variedade de diálogos homem-computador e um entendimento substancial desta comunicação.

Zwicker & Reinhard (1990) alertam para a relação custo/benefício envolvida na incorporação de inteligência nas interfaces. Segundo estes autores, o custo é alto e o benefício pode ser mínimo ou nenhum podendo ter, ao contrário, um efeito complicador na interação. Portanto, interfaces inteligentes são recursos certamente úteis mas devem ser aplicadas a um conjunto limitado de contextos, em especial aos sistemas realmente complexos.



## CAPITULO 3

### ASPECTOS IMPORTANTES NO DESENVOLVIMENTO DE INTERFACES

#### 3.1 Introdução

O processo de desenvolvimento de interfaces homem-computador pode ser visto como um caso específico do processo de desenvolvimento de sistemas de aplicação computacional. Conforme mostra o modelo da figura 3.1, sugerido por Rubistein & Hersh (1984), o processo de desenvolvimento de sistemas pode ser dividido nas etapas coleta de informação, projeto do sistema, implementação, avaliação e entrega do sistema à comunidade usuária. Neste modelo a informação externa é coletada e usada para estabelecer objetivos, limites e outros requerimentos de projeto. O sistema é projetado, construído e testado de forma interativa e iterativa até satisfazer algum critério, quando, então, o sistema é liberado. A informação sobre a aceitação do mercado e reações dos usuários pode ser usada como entrada no próximo ciclo do desenvolvimento de sistemas. Cada uma das etapas pode ser decomposta em componentes e atividades, conforme descrito a seguir.

A coleta de informação é o ponto de partida e o ponto de referência no processo de desenvolvimento do projeto. A principal atividade desta fase é a especificação dos requerimentos, que consiste do desenvolvimento de um entendimento do domínio do problema com subsequente documentação. Estando os requerimentos e objetivos explicitados, estes devem ser utilizados como restrições para orientar o projeto. Esta informação deve ser utilizada também para decisão entre os inevitáveis e conflitantes compromissos de projeto. Uma variedade de fontes de informação fornece a base para o projeto de sistemas de computação. Estas fontes incluem requerimentos técnicos, estado-da-arte corrente, produtos similares já em uso, padrões industriais, leis governamentais e as capacidades técnicas dos projetistas do

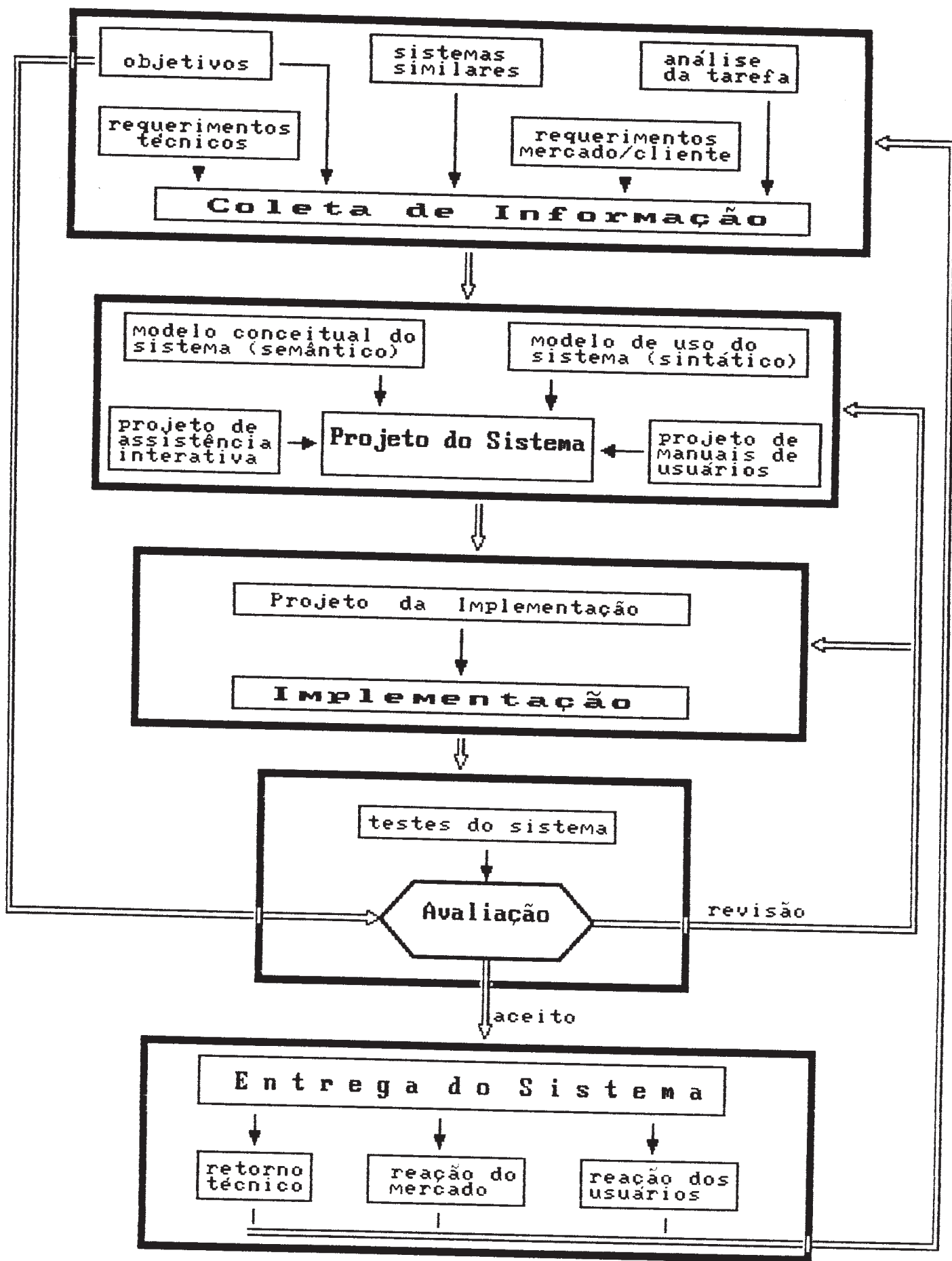


Figura 3.1 Processo de Desenvolvimento de Interfaces

sistema. Descrições detalhadas dos requerimentos do mercado ou de um cliente específico juntamente com a análise das tarefas que o usuário deverá desempenhar ajudam a fornecer uma idéia clara dos usuários e seu ambiente. Por último, é necessário uma percepção nítida dos objetivos do projeto e de sua importância relativa. Os objetivos do processo tais como o tempo de desenvolvimento permitido, os recursos disponíveis e o orçamento para o projeto são também considerações importantes. A análise de toda esta informação juntamente com os objetivos formam os limites que vão guiar o projeto.

A informação coletada e organizada, juntamente com os objetivos do projeto identificados, dão ao projetista uma idéia clara dos requerimentos e restrições do sistema a ser criado e formam a semente a partir da qual o projeto crescerá. Uma descrição formal do sistema deve especificar as funções que o sistema irá desempenhar, como os usuários utilizarão o sistema, e qual o papel do sistema no seu trabalho. Para isso são desenvolvidos os modelos semântico (conceitual) e sintático (de uso) do sistema, que devem ser baseados nos estudos dos modelos mentais dos usuários para a aplicação em questão, conforme descrito no capítulo 2. Os modelos semântico e sintático são especificados, a assistência interativa e os manuais de usuários são projetados e o conjunto é composto para formar o projeto do sistema.

O projeto de um sistema é o processo de especificação do comportamento externo do sistema - os detalhes de software e hardware que impactam o usuário. Neste estágio é essencial desenvolver a documentação e, paralelamente, o projeto de assistência interativa. O processo de documentação oferece uma oportunidade para avaliar o sistema como um todo antes que a primeira linha de código seja escrita. A aptidão para escrever um manual de usuário coerente é sinal que o projeto está consistente e claro. Os modelos semântico e sintático, os projetos de manuais de usuários e de assistência interativa são compostos na documentação que forma o projeto do sistema.

A criação de um projeto completo de implementação antes do início da codificação é uma boa prática de engenharia de software

para sistemas de todos os tamanhos. Esta prática é uma ferramenta básica para prevenir situações indesejadas. No entanto, não importa quão cuidadoso o projeto tenha sido feito, a implementação sempre descobre situações não especificadas e questões que necessitam de solução. A alternativa é construir protótipos. Os protótipos podem ser construídos a um custo menor do que o custo de sistemas completos e em menos tempo. Os protótipos são então analisados e iterativamente refinados. Esta alternativa é alentada pelo fato de que o presente estado das pesquisas em fatores humanos não permite a obtenção da "interface certa" na primeira tentativa.

Protótipos permitem a observação do comportamento da interface o mais cedo possível e a modificação fácil dos projetos. O protótipo reduz as possibilidades de surpresas para o usuário final, ajuda a resolver o problema da inabilidade do usuário em fornecer especificações completas para os projetistas do sistema e dá a este um sentimento mais imediato do sistema proposto. Ele revela desentendimentos que surgem entre os projetistas e os usuários finais devido às suas diferentes formações e experiências. Prototipagem rápida (Hartson & Hix, 1989) permite que o processo de refinamento iterativo ocorra mais cedo no processo de projeto e é uma forma eficiente e efetiva de envolver o usuário final no projeto do sistema.

A etapa de avaliação possibilita um processo de realimentação iterativo que auxilia o projetista a melhorar o sistema de forma organizada, com base nos objetivos de projeto. Um projeto de avaliação deve ser elaborado, os testes devem ser feitos com base neste projeto e os resultados avaliados. Embora testes, verificação e validação tenham como objetivo indicar se um projeto preenche os requerimentos especificados, a prototipagem pode mostrar erros nos requerimentos. A comunicação rápida entre fomentadores, usuários finais e implementadores fornece conclusões que realimentam o projeto do sistema e a implementação. Usuários de sistemas desenvolvidos via protótipo são, em geral, mais favoráveis ao sistema final do que usuários de sistemas que não foram prototipados (Hartson & Hix, 1989).

O processo de projeto é completado com a entrega do sistema

aos usuários. Após a entrega ainda existem muitas perguntas e muita informação a ser recolhida em relação ao uso do sistema, à satisfação dos usuários, à aceitação ou não do mercado e porquê. Esta informação pode ser usada no desenvolvimento de um novo projeto.

Dentre os aspectos do processo de desenvolvimento de interfaces citados nesta introdução, quatro são abordados de forma mais detalhada nas seções subsequentes: independência do diálogo, assistência ao usuário, apresentação da informação na tela e avaliação de interfaces.

### **3.2 Independência do Diálogo**

Historicamente, os sistemas interativos têm sido projetados com os componentes do diálogo homem-computador e da aplicação computacional intercalados. O problema desta abordagem é que à medida que o projeto cresce, torna-se difícil manter e modificar o sistema. O desenvolvimento de interfaces homem-computador com qualidade envolve um ciclo iterativo de projeto e avaliação e, portanto, um importante critério no desenvolvimento de interfaces é a possibilidade de modificação fácil e rápida. Para isso é necessário algum tipo de separação entre estes dois componentes.

O conceito de independência do diálogo (Coutaz, 1985; Draper & Norman, 1985) foi desenvolvido a partir do conceito de modularização (Myers, 1978) emprestado da engenharia de software. Independência do diálogo consiste em agrupar a aplicação computacional em um conjunto de módulos e a interface com o usuário em outro conjunto de módulos e elaborar uma estrutura formal para comunicação entre os dois conjuntos de módulos formados. Portanto, cada conjunto de módulos pode ser construído e modificado de maneira independente, sem afetar o resto do sistema. Na prática isto significa que a aparência da interface e os métodos utilizados para extrair dados de entrada do usuário não são conhecidos pelo programa de aplicação, ou seja, a única parte do sistema que interage com o usuário é o módulo da interface. A Figura 3.2 mostra um esquema simplificado da interação homem-computador ressaltando os módulos da interface e

da aplicação.

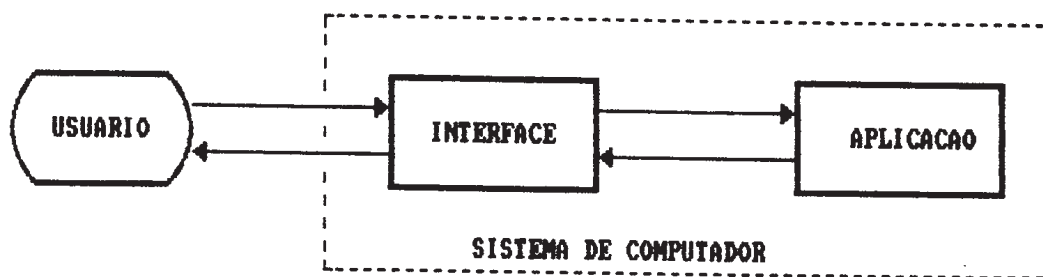


Figura 3.2 Componentes da Interação Usuário-Computador

Três razões justificam e motivam o projeto considerando independência do diálogo, a saber:

1. Flexibilidade do projeto da interface e da própria aplicação. Neste contexto, dois problemas diferentes estão envolvidos: um problema de ciência da computação e outro de fatores humanos (Hartson & Hix, 1989). No problema de ciência da computação cada rotina necessita de um conjunto de entradas válidas. A função computacional não cuida de como os valores de entrada são fornecidos pelo usuário, de como eles são validados, ou mesmo de onde vêm. Por outro lado, o problema relativo aos fatores humanos é projetar um diálogo que "transforme" o computador em uma ferramenta que auxilie o usuário a desempenhar sua tarefa. Desta perspectiva o diálogo deve ser fácil de usar, deve ser receptivo aos valores de entrada fornecidos pelo usuário, deve fornecer realimentação e esclarecimentos quando necessário, deve confrontar os valores de entrada contra os critérios de validação e guiar o usuário de entradas que não satisfazem o critério para entradas válidas. As entradas válidas são, então, enviadas ao componente computacional. Com a independência do diálogo, soluções separadas para os dois problemas podem ser geradas, modificadas ou expandidas

de forma independente.

2. Em função da separação em dois problemas, cada módulo pode ser projetado por um especialista na área. A independência do diálogo pode criar um novo papel no projeto de um sistema que é o do *projetista do diálogo* (Hartson & Hix, 1989; Monk, 1984; Shneiderman, 1987), um especialista em fatores humanos envolvido com o projeto, implementação e avaliação da forma, estilo, conteúdo e sequenciamento das interfaces homem-computador. O projetista do diálogo deve ser sensível às necessidades cognitivas do usuário final. Durante o projeto e a implementação do diálogo, este especialista deve utilizar princípios de fatores humanos para construir, avaliar e refinar interativamente a interface. Independência do diálogo permite que modificações sejam feitas rapidamente, possibilitando um ciclo de avaliação e revisão.
3. Devido a esta modularização, uma interface pode ser compartilhada por diversas aplicações. Conseqüentemente, o esforço e o custo de desenvolvimento da interface ocorrem apenas uma vez e os usuários podem interagir com muitos programas de aplicação de forma consistente através da mesma interface. Apenas um modelo conceitual é construído e esse é utilizado para operar as aplicações da estação de trabalho. O reverso consiste em que, com a independência do diálogo, diferentes interfaces podem compartilhar o mesmo componente computacional, adequando a aplicação aos diversos tipos de usuário.

Independência do diálogo é suportada por uma separação em tempo de projeto do diálogo e da aplicação. Isto significa que um sistema interativo deve ser composto de um componente do diálogo, através do qual toda comunicação entre usuário final e sistema acontece, e um componente computacional, com a funcionalidade adequada ao domínio da tarefa de interesse do usuário.

### **3.3 Assistência ao Usuário**

A necessidade de sistemas de computador e programas interativos auto explicativos, isto é, sistemas ou programas que

instruam o usuário sobre seu próprio uso e forneçam um ambiente amigável e conveniente de trabalho, está surgindo como uma questão significativa no projeto de sistemas interativos.

A interface com o usuário pode ser limitada como qualquer parte de um sistema com a qual o usuário estabelece contato físico, perceptivo ou conceitual. Qualquer aspecto do sistema que pertença ao modelo mental do usuário é parte da interface. Portanto, a interface inclui toda a comunicação com o usuário e, conseqüentemente, toda a parte de assistência ao usuário.

O objetivo geral e mais importante de um sistema de auxílio ao usuário em um ambiente interativo é fornecer assistência de maneira consistente em todos os estágios do sistema. É desejável para o usuário estar apto a requisitar assistência sempre que tiver dúvida sobre como proceder, e esperar uma resposta que relaciona e explica as opções de entrada disponíveis. Outra facilidade desejável consiste em prover o sistema com boas mensagens de erro e com opções de recuperação para utilizar, sempre que um erro for detectado.

Para implementar um sistema de ajuda eficiente, deve existir não só uma base de informação de ajuda sobre a qual o sistema é construído, como também disponibilidade de acesso constante à rotina de ajuda. Este tipo de acesso implica que a interface deve ter flexibilidade para permitir ao usuário sair temporariamente do ambiente de processamento do comando corrente e entrar no ambiente de assistência ao usuário. O ambiente de assistência ao usuário deve ter acesso à informação suficiente no ambiente de processamento do comando para fornecer ajuda diretamente, no contexto em que o usuário está operando e no tempo em que acontece um erro ou uma requisição de ajuda.

Um sistema de ajuda deve fornecer informação interativamente, assim como ter capacidade de produzir manuais de usuário impressos da forma convencional. Alguns requerimentos são necessários para formar uma base de informação de ajuda. Segundo Fenchel (1980), um sistema de auxílio deve:

1. De forma consistente e em todos os pontos de entrada, fornecer uma lista de opções disponíveis de entrada para o



usuário no estágio corrente do sistema; no entanto, o programa de aplicação deve ser projetado de forma que o número de opções não seja muito grande.

2. Descrever cada opção de entrada com o objetivo de permitir ao usuário escolher inteligentemente que opção é desejada. As descrições devem ser concisas e os usuários devem ter a opção de requisitar detalhes adicionais.
3. Apresentar e descrever o contexto do comando corrente. Os usuários devem estar aptos para determinar a função de um conjunto de entradas bem como o significado de cada entrada individual.
4. Descrever um erro ocorrido apresentando (a) razão para sua ocorrência, (b) consequência do erro, isto é, estado resultante do sistema e (c) opções para recuperação como, por exemplo, continuar a entrada ou reiniciar o comando. Informação sobre o erro deve ser concisa, embora permitindo ao usuário requisitar informação adicional.
5. Fornecer uma descrição geral do uso do sistema e, em particular, um tutorial sobre o uso do sistema de auxílio. Os usuários devem entender a funcionalidade geral de um sistema e os conceitos básicos de seu uso. Sistemas de ajuda são muito menos efetivos se seus usuários não têm algum entendimento básico dos conceitos do sistema.

Um sistema de ajuda integral é um sistema de ajuda que é projetado e implementado como parte do projeto de um sistema interativo como um todo. Se o projeto e a implementação mudam, o sistema de ajuda também deve mudar. Se um sistema de auxílio for incorporado como um requerimento de projeto de um sistema interativo, então ele poderá manter a informação atualizada e precisa, poderá compartilhar informação de projeto e poderá ter geração automática de mensagens de erro cobrindo todos os possíveis erros dos usuários, o que é difícil de se obter de outra forma. Além disto, quando o sistema de auxílio é projetado ao mesmo tempo que o sistema que ele suporta, o tipo de informação a ser gerado é considerado por todo o processo de projeto. É muito mais difícil gerar e manter informação de ajuda

posteriormente ao projeto do sistema.

A informação de assistência ao usuário pode ser dividida, para propósitos de discussão, em quatro partes, a saber:

1. assistência para uso geral do sistema;
2. assistência para partes específicas do sistema;
3. assistência para erros;
4. apresentação da assistência ao usuário.

Divisões de auxílio sintático e semântico são aplicáveis a todas as categorias.

### **Assistência para uso geral do sistema**

Usuários de sistemas interativos necessitam ter um entendimento geral da funcionalidade do sistema em uso. A faixa de funções disponíveis deve ser entendida com ênfase para aquelas comumente usadas e para as funções essenciais. Além do entendimento do domínio do sistema, os usuários necessitam entender a essência do sistema, isto é, as características de projeto que são preservadas na maior parte do sistema. Em sistemas com auxílio integral é fundamental que o usuário entenda que a ajuda está constantemente disponível e que saiba como requisitá-la quando necessário. O sistema pode ser apresentado aos usuários sob a forma de um conjunto de tutoriais interativos ou sob a forma de um exemplo de uso do sistema.

### **Assistência para partes específicas do sistema**

Quando os usuários já têm uma idéia geral do sistema, eles necessitam de informação sobre dispositivos específicos e comandos disponíveis no sistema. A informação requerida inclui a sintaxe e a semântica do comando. A informação semântica deve incluir uma descrição da função e do efeito do comando, um exemplo de uso do comando e notas contendo exceções relacionadas com o uso ou com o significado do comando. Além disto, a assistência para comandos deve incluir quaisquer sinônimos para o comando e, quando aplicáveis, sugestões sobre como e porque usar o comando.

## **Assistência para erros**

Usuários frequentemente erram devido a dois fatores: falta de entendimento da operação do sistema ou de algum aspecto do sistema, e dificuldades na entrada de dados e outras falhas. O sistema de ajuda deve assistir os usuários em ambos os tipos de erro. Se um erro for causado por falta de entendimento, então o usuário requer assistência na forma de uma explanação da causa do erro e de sua severidade em termos de qualquer efeito colateral adverso causado pelo erro. Além disto, o usuário deve ser ajudado na correção do erro e na continuação do uso do sistema. Quanto ao segundo tipo de erro, o sistema deve ser projetado para minimizar sua ocorrência sendo tolerante a erros simples de grafia e formatação.

Um sistema interativo deve responder aos erros com uma mensagem descritiva e concisa indicando que um erro ocorreu e indicando que parte da entrada causou o erro. Sempre que possível os efeitos do erro devem ser localizados, isto é, o erro deve ser detectado logo que seja possível e não deve afetar outros comandos válidos do usuário. Sistemas que oferecem operações para desfazer os efeitos indesejados de um comando errôneo são preferíveis.

## **Apresentação da assistência ao usuário**

A qualidade da apresentação da informação de assistência ao usuário é tão importante quanto a informação que lhe é apresentada. A apresentação pode e deve ser de diversas formas. Assistência ao usuário deve estar disponível em linha durante o uso de um sistema interativo, a qualquer ponto no qual o usuário possa fornecer entrada para o sistema.

A qualidade da informação apresentada para cada requisição é um fator crucial. Muita ou muito pouca informação não é desejável. O ideal é apresentar uma mensagem concisa para a requisição inicial de assistência, seguida por respostas progressivamente maiores às requisições adicionais. É desejável que o usuário possa suprimir a apresentação padrão e requisitar

explicitamente o tipo e a quantidade de informação de assistência a ser gerada.

Informação de assistência global deve estar disponível através de programas tutoriais projetados para novos usuários. O manual de usuário convencional é um componente importante de um sistema interativo, porque é importante para os usuários a possibilidade de ler uma descrição da operação e função de um sistema e seus comandos enquanto não estão utilizando o sistema.

### **3.4 Apresentação da Informação na Tela**

A parte mais visível do projeto de interfaces homem-computador é a que diz respeito à apresentação dos dados no terminal de vídeo de um sistema de computação. A apresentação dos dados na tela tem evoluído da forma estática e linear, representada pelas intermináveis listas sequenciais de dados, para uma forma rica e graficamente complexa que transmite informação compactada e com capacidade para mudar o caráter da troca de informação homem-computador. A interface é a única visão que o usuário tem do sistema, e o usuário estende sua impressão sobre a interface para o próprio sistema.

A forma de apresentação dos dados define padrões e significados que influenciam na maneira pela qual o usuário vê e interpreta a tela, as janelas, a informação e os relacionamentos físicos e semânticos entre os grupamentos de informação. Para obter o efeito desejado os projetistas usam janelas múltiplas, cores, menus, ícones e outras técnicas visuais com o objetivo de aumentar a efetividade na interação, transmitir um sentido de como a superfície total da tela deve ser interpretada, além de acordar o usuário e sustentar seu envolvimento durante a interação. O projeto físico engloba tudo o que é visto na tela estaticamente e dinamicamente.

A atenção para o caráter estético (beleza) é uma questão importante para tornar as interfaces inteligíveis, recordativas e atraentes para os usuários. Conseqüentemente, o caráter estético é um fator de satisfação para os usuários. A parte exibida da interface pode ser vista como um projeto visual onde os aspectos

técnicos são aplicados visando resultados artísticos. Medidas objetivas e precisas de excelência para projetos visuais de interfaces são difíceis de serem obtidas porque, na verdade, cada projeto visual de interface baseia-se essencialmente em critérios subjetivos.

A importância do projeto visual no planejamento de interfaces com o usuário é confirmada por Croucher (1986), Norman, Weldon & Shneiderman, (1986) e Shneiderman (1987), que apontam as seguintes variáveis relevantes neste nível do projeto:

1. uso das cores
2. velocidade de apresentação da informação
3. quantidade de informação na tela
4. paginação e rolagem de telas
5. projeto das janelas

Para cada uma das variáveis citadas existem linhas gerais de projeto baseadas em resultados de experimentos realizados e descritos por Croucher (1986), Gait (1986) e Shneiderman (1987). A seguir cada uma destas variáveis será abordada com descrição de seu efeito e parâmetros de uso recomendados pelos resultados empíricos descritos pelos autores citados.

### **Uso das Cores**

Estudos sobre cores mostram que, devido à variações de luminescência, nem todas as cores de um texto exibido em terminal de vídeo são igualmente legíveis, conforme mostra a Tabela 3.1. Além disto, Croucher (1986) apresenta estudos que demonstram que o olho humano é maximamente sensível aos comprimentos de onda na área amarelo/verde do espectro de cores, e que o vermelho e o azul são significativamente menos brilhantes ao serem comparados com as demais cores. Com base nestes estudos, Croucher (1986) concluiu que o branco, o amarelo, o ciano e o verde são, nesta ordem, as cores mais fáceis de ler em telas luminosas, e que o vermelho, o azul e talvez o magenta devem ser evitados. A Tabela 3.2 mostra o resultado de uma pesquisa onde os usuários deram notas de valor máximo 10 para cada tela a eles apresentada. Os resultados confirmam a conclusão acima.

**TABELA 3.1**

**Luminescência Relativa das Cores (Fonte: Croucher, 1986)**

---

Cor	Luminescência Relativa
branco	100
amarelo	89
ciano	70
verde	59
magenta	41
vermelho	30
azul	11

---

**TABELA 3.2**

**Preferência por Textos Coloridos sobre Fundo Preto  
(Fonte: Croucher, 1986)**

---

Cor do texto	Média obtida (máximo = 10)
ciano	7.36
verde	7.26
amarelo	7.12
branco	6.88
magenta	5.63
vermelho	4.69
azul	3.23

---

Estas conclusões foram encontradas assumindo-se que todo texto colorido foi apresentado em fundo preto. Outras cores de fundo foram pesquisadas em uma experiência na qual a cada usuário foi apresentada uma tela cheia de texto colorido mostrado sobre diferentes cores de fundo. Os usuários atribuíram notas de no máximo 10 a cada página, como anteriormente. Para o propósito deste teste, ciano, amarelo, branco e verde foram consideradas cores claras, e azul, vermelho e magenta foram consideradas cores escuras. Os resultados são mostrados na Tabela 3.3. Texto de cor clara sobre fundo claro não foi testado, porque obviamente é difícil de ler. Pode-se observar que cores contrastantes tiveram melhor desempenho que cores similares e que cores claras sobre fundos escuros foram preferidas em relação às demais combinações.

**TABELA 3.3**

**Preferência por Textos Coloridos sobre Fundos Coloridos  
(Fonte: Croucher, 1986)**

Cores	Categoria	Média
amarelo sobre azul	claro sobre escuro	7.00
ciano sobre azul	claro sobre escuro	6.66
branco sobre vermelho	claro sobre escuro	6.51
vermelho sobre branco	escuro sobre claro	4.66
azul sobre ciano	escuro sobre claro	4.65
azul sobre amarelo	escuro sobre claro	4.15
azul sobre magenta	escuro sobre claro	3.41
verde sobre magenta	claro sobre escuro	3.36

Shneiderman (1987) sugere diversas regras para o uso de cores, entre elas:

1. Limitar o número de cores em uma única tela a 4 (quatro), com um total de 7 (sete) cores para a sequência completa de telas.

2. Reconhecer a potência das cores como técnica de codificação. O projetista deve conhecer o código de cores válido no domínio da tarefa considerada pela aplicação com a qual está trabalhando. Por exemplo, vermelho pode significar saldo negativo em uma aplicação bancária, temperatura elevada em uma aplicação de engenharia e perigo em assuntos de segurança. O esquema de codificação utilizado pelo projetista deve ser de fácil interpretação e, quando apropriado, o código de cores deve ser colocado na tela ou em uma janela auxiliar. As cores aceleram o reconhecimento de muitas tarefas mas podem inibir o desempenho de tarefas que vão contra o esquema de codificação.
3. As cores podem ajudar na formatação de telas lotadas de informação, onde o espaço é precioso. Cores similares podem ser usadas para agrupar itens relacionados e cores diferentes podem ser usadas para distinguir campos que estão fisicamente perto mas que são logicamente distintos. O aninhamento de níveis pode ser mostrado através de uma progressão de cores como, por exemplo, vermelho, alaranjado, amarelo e branco.
4. A codificação de cores deve ser consistente em todo o sistema. Se, por exemplo, as mensagens de erro estão em vermelho, o projetista deve certificar-se de que todas as mensagens de erro aparecem em vermelho. Uma mudança de cor pode ser interpretada como uma mudança na importância da mensagem.
5. As mudanças de cores podem ser usadas para indicar mudanças de estado. Desta forma as cores agem como um método para obter a atenção do usuário. Este método é especialmente útil quando centenas de valores devem ser continuamente mostrados na tela.

### **Velocidade de Apresentação da Informação**

Croucher (1986) relata experiências onde a informação foi apresentada como texto escrito na tela com uma pausa entre a



geração de cada caractere. Uma pausa de 0,4 segundo entre cada caractere apresenta texto na velocidade de leitura média. Durante a execução das tarefas esta taxa variou para permitir velocidades de apresentação maiores e menores. O método alternativo de geração de texto é eliminar qualquer pausa entre a geração de caracteres e apresentar uma tela inteira de texto de uma só vez, porém permitindo ao usuário sua leitura antes de prosseguir para a próxima tela. Geração de texto pausada, caractere a caractere, foi preferida à apresentação em bloco por 65% dos usuários pesquisados, que a consideraram mais fácil de ler desde que a apresentação não seja significativamente mais lenta que a velocidade de leitura média.

### **Quantidade de Informação na Tela**

Os experimentos mostraram que uma tela demasiadamente cheia de dados produz mais erros e resulta em tempos de exploração mais longos. Uma tela cheia de informação é mais difícil de digerir que uma série de telas curtas com informação centrada e espaçada.

### **Paginação e Rolagem de Tela**

Usuários testaram a rolagem de tela como alternativa à paginação. A paginação foi preferida à rolagem de tela. No caso de se usar rolagem de tela, o usuário deve ter a facilidade para parar a página a qualquer tempo. Ao usuário deve ser permitido sentir-se no controle da apresentação de informação e estar apto para parar uma série de telas ou um fluxo de texto a qualquer tempo.

### **Projeto das Janelas**

Janelas servem como um ambiente conceitual para capturar, desenvolver, organizar e realçar informação textual ou gráfica. Janelas bonitas (*pretty windows*, Gait, 1986) é uma das diversas áreas onde considerações estéticas podem valorizar e aperfeiçoar a interface exibida.

Uma janela bonita é uma área na tela que é um quadrado ou um retângulo áureo. Retângulo áureo é todo retângulo no qual a razão de sua altura e largura constitui a razão áurea de Euclides: aproximadamente 1,618.

Retângulos áureos têm sido usados há muitos anos por profissionais responsáveis pela criação de objetos visualmente agradáveis, tais como artistas, arquitetos, projetistas, psicólogos e publicitários. A preferência por retângulos áureos pode ser considerada estética e subjetiva e tem fundamentos empíricos em experimentos psicológicos, conforme descrito em Gait (1986). De acordo com esta descoberta, interfaces que utilizam janelas bonitas podem desfrutar de uma vantagem ergonômica cognitiva, com o ambiente conduzindo o usuário a uma maior produtividade através da maximização do potencial de comunicação do display.

Ambientes de interação com janelas são quase lugares-comuns em sistemas de computação modernos. Muitas vezes é necessário dividir uma janela ou uma tela inteira em partes. Estas partições têm os objetivos de (1) organizar a informação para uso conveniente, (2) fornecer padrões para as partições com o objetivo de aliviar o programador de detalhes, (3) fazer ótimo uso da área da tela e (4) fornecer um modelo de projeto de tela para guiar o programador no controle de sua aparência. O relacionamento geométrico natural entre os quadrados e os retângulos áureos durante a partição destes, conforme mostra a Figura 3.3, unifica os conceitos de janela bonita e partição bonita.

Dado o número de partições desejado em uma janela, qualquer janela bonita pode ser partida em no máximo dois quadrados e demais retângulos áureos quantas forem as partições requeridas pelo usuário. As definições formais destes conceitos e as provas matemáticas da partição estão fora do objetivo deste trabalho mas podem ser encontradas em Gait (1986).

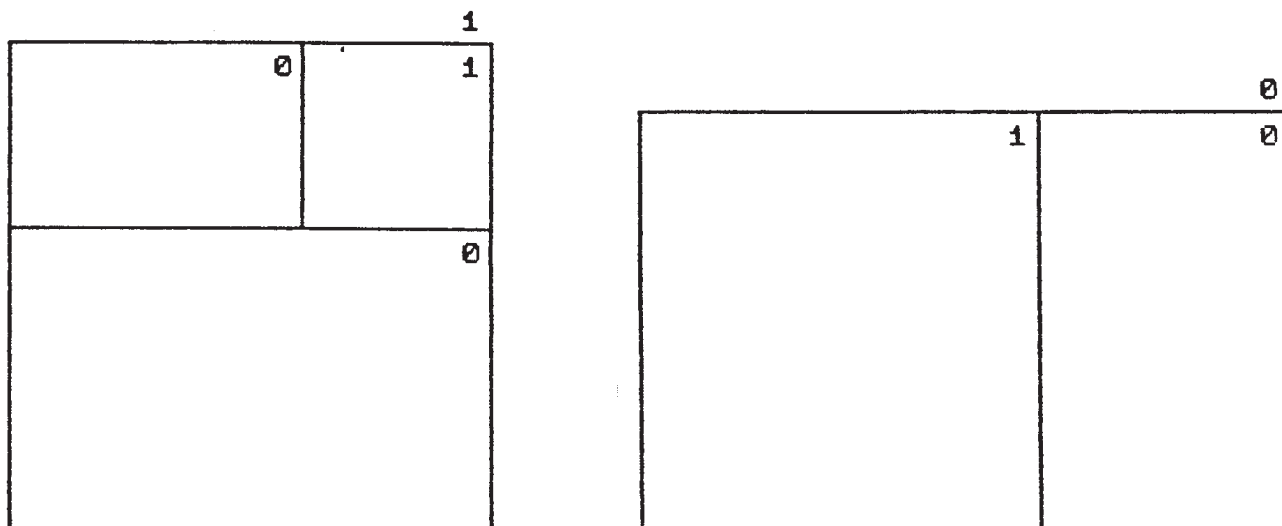


Figura 3.3 Janelas bonitas com partições bonitas.

A figura 3.3 apresenta exemplos de janelas com partições bonitas. O retângulo áureo à direita da figura foi dividido em um quadrado e um subretângulo áureo. O quadrado à esquerda foi dividido em um subquadrado e dois retângulos áureos. Todas as partições de janelas bonitas são geradas a partir destas duas partições.

### 3.5 AVALIAÇÃO

A avaliação de interfaces homem-computador tem como objetivos:

1. verificar se os objetivos do projeto foram cumpridos;
2. analisar a interface como um todo, identificando as áreas problemáticas;
3. depurar a interface.

A única maneira de avaliar interfaces é observar e colher dados sobre sua utilização e analisar estes dados incorporando os resultados e conclusões obtidas na própria interface. Os dados de interesse se referem a quais partes da interface apresentam problemas, quais são as telas e os comandos mais utilizados,

quais são as operações que confundem o usuário e quais são as operações com maior índice de erros.

Shneiderman (1987) sugere que a especificação de uma interface seja feita em termos de um conjunto de testes de aceitação, com resultados possíveis para tempos de resposta e com requerimentos de software e hardware especificados. Se o produto final falha nestes testes, o sistema deve ser melhorado até a obtenção do sucesso nos testes. Para isto, a especificação de uma interface deve cobrir, no mínimo, os seguintes pontos:

1. descrição dos usuários-alvo e escolha de usuários representativos para participarem dos testes;
2. descrição das tarefas a serem desempenhadas e das circunstâncias nas quais elas serão executadas;
3. especificação das medidas de interesse e dos valores a serem encontrados para cada medida; se o teste é para nova versão de um sistema, a especificação de valores é fácil; caso contrário, o estabelecimento de valores apropriados só é possível através de processo interativo. As medidas de interesse são:
  - tempo de aprendizado para funções específicas
  - velocidade de desempenho de tarefas específicas
  - taxa de erros
  - satisfação subjetiva do usuário
  - retenção de comandos no tempo
  - número de requisições de ajuda para cumprir determinada tarefa
  - atitude do usuário.

Um exemplo de teste de aceitação é o seguinte:

Trinta usuários típicos serão treinados no uso do sistema por 45 minutos. Estes usuários terão 15 minutos para executar uma bateria de tarefas. A taxa média de execução das tarefas deve ser acima de 80% e o número médio de erros deve ser menor que 3.

Estudiosos e projetistas de interfaces consideram os testes interativos com usuários um fator essencial na construção de interfaces fáceis de usar. Durante os testes os usuários podem fazer sugestões que irão ajudar a adaptar o sistema às suas necessidades, e podem indicar o que eles não gostam. Perlman (1989) afirma que o projetista pode colher dados conversando com especialistas em fatores humanos, projetos gráficos e outros envolvidos no projeto de interfaces, entrevistando clientes, aplicando questionários, desenvolvendo protocolos de utilização do sistema ou ainda instrumentando o programa para controlar sua utilização. O custo de obtenção destes dados pode ser alto se os experimentos forem cuidadosamente controlados para obtenção de decisões conclusivas de projeto, mas experiências de custo baixo para colher dados também são possíveis.

Fazer o protótipo de uma interface o mais cedo possível permite testes mais extensos. Os testes com o protótipo irão expor os erros da interface e do sistema. Testes interativos fornecem também grande credibilidade ao projeto. Descobrir reações negativas dos usuários durante o desenvolvimento do sistema é melhor do que descobri-las após sua liberação final. Quando os usuários são convidados a participar dos testes, eles ficam mais envolvidos no projeto do sistema e gostam de ver suas sugestões incorporadas na versão final. Os testes devem ser feitos também em manuais e mensagens de erro.

Maedo, Miyake, Nievergelt & Saito (1984) desenvolveram uma lista de pontos úteis para avaliar rapidamente a qualidade geral de uma interface homem-computador, sem considerar todos os detalhes do projeto. Segundo estes autores, alguns pontos devem ser observados na descrição de uma interface homem-computador ou em uma avaliação rápida de sua adequação para uma dada classe de usuários ou tarefas. Estes pontos são considerações-chave que determinam a qualidade da interface e formam uma lista pequena, sem intenção de cobrir todos os pontos relevantes, mas com o objetivo de ser funcional. A lista é composta de seis itens estruturados da seguinte maneira:

## **Classe de Usuários**

Deve ser considerada juntamente com a tarefa que os usuários querem cumprir usando o sistema. Para a avaliação, deve-se listar a classe de usuários e as tarefas a serem cumpridas.

## **Conjunto de Capacidades Funcionais**

As capacidades funcionais determinam o que o sistema pode fazer. A interface física e a interface lógica determinam como o usuário invoca as capacidades do sistema. Existe um forte relacionamento entre as capacidades de um sistema e sua interface no sentido de que certas funções tornam-se impraticáveis ou mesmo impossíveis se apresentadas através de uma interface inadequada.

Para a avaliação devem ser listados os comandos e suas funções. Os comandos podem ser classificados em categorias como, por exemplo, comandos de consulta, de manipulação de dados, utilitários, de controle de diálogo, de ajuda e de comunicação com outros sistemas.

## **Interface Física**

Neste item devem ser listados os dispositivos de entrada e de saída.

## **Interface Lógica**

A interface lógica é o modelo conceitual do sistema. Este modelo diz respeito aos objetos que o usuário pode manipular, operações entre estes objetos e os comandos utilizados para realizar estas operações. Na interface lógica devem ser analisados quais são os conceitos básicos, estruturas e operações, ou melhor, qual é o modelo conceitual no qual o sistema está baseado.

## **Realização Física**

Mostra como a interface lógica do sistema está mapeada na interface física, ou seja, como as operações são ativadas.

### **Interação Sistema-Usuário**

Como o sistema como um todo (capacidades funcionais, interfaces física e lógica) corresponde às classes de usuários e suas tarefas. Este item pode ser subdividido em quatro outros:

1. Correspondência entre o sistema e o conhecimento e habilidades do usuário. Diz respeito ao nome dos comandos e seus conceitos e ao desenho do teclado e projeto de tela.
2. Projeto para erros humanos. Mede a tolerância ao erro e a detecção de erro. Se é possível reverter comandos, se o sistema está protegido dos erros dos usuários e se o sistema é capaz de detectar erros mais óbvios e sugerir correções.
3. Projeto para aprendizado e personalização. Avalia as ajudas interativas que o sistema fornece, a adaptação que o sistema tem em relação ao nível de habilidade dos usuários (novatos, experientes, etc.), as mensagens e a habilidade de personalização que o sistema permite.
4. Projeto para memória humana. Considera fatores como se o sistema mostra na tela, sempre que desejado, seu estado corrente, se o sistema tem auxílio para usuários iniciantes ou intermitentes e se o sistema tem sempre realimentação perceptível às entradas dos usuários.

A figura 3.4 mostra a ligação entre estes conceitos.

Como efeito colateral, a avaliação de interfaces poderá servir para conscientizar projetistas e pesquisadores de que

1. interfaces homem-máquina são produtos para ser discutidos, comparados e criticados tal como, por exemplo, são as linguagens de programação,
2. os sistemas atualmente diferem amplamente na qualidade da interface que eles apresentam ao usuário e esta qualidade afeta a produtividade do usuário e
3. a especificação funcional de um sistema interativo deve começar com a especificação do modelo que o usuário tem do sistema.

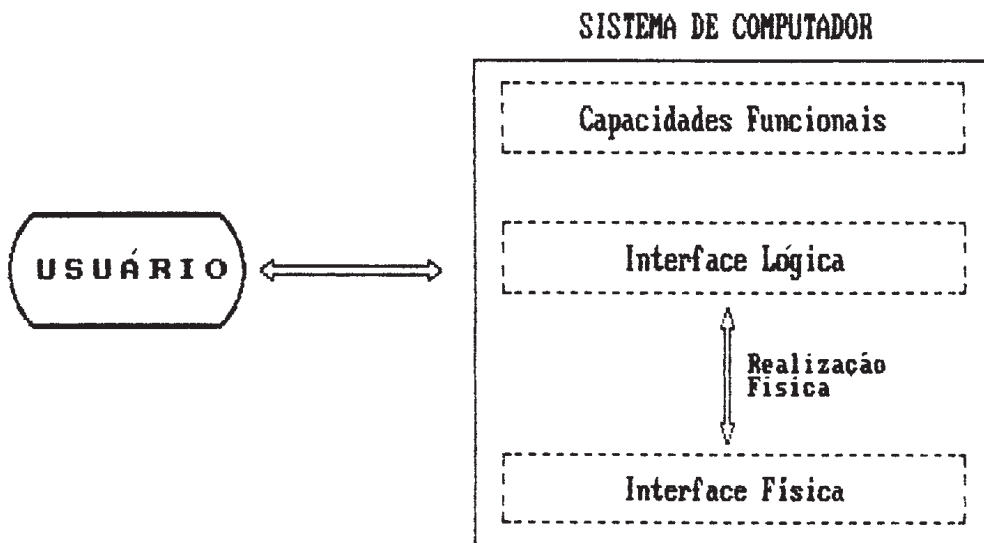


Figura 3.4 Interação Sistema-Usuário



## CAPITULO 4

### ESPECIFICAÇÃO DE UMA INTERFACE PARA O SISTEMA PATPLUS

#### 4.1 O Sistema PATPLUS

##### 4.1.1 Descrição do Sistema

O PATPLUS é um sistema de recuperação de informação em textos projetado para dar a seu usuário flexibilidade total quanto à estruturação dos dados. Está, portanto, incluído na categoria dos sistemas que permitem que seus usuários criem e pesquisem seus próprios bancos de dados em microcomputadores, sem as limitações, a complexidade e o alto custo dos sistemas voltados para computadores de grande porte. Tem aplicação tanto em organizações pequenas, como escritórios de forma geral, quanto em organizações grandes, onde cada uma de suas divisões e subdivisões poderá organizar e pesquisar seu próprio material. Pode ser utilizado também em bibliotecas, instituições acadêmicas e em bancos de dados de uso pessoal. O texto de entrada para o PATPLUS é qualquer arquivo digitado em um editor de textos que respeite a convenção de caracteres ASCII, não havendo nenhuma restrição quanto à estruturação dos dados. Este texto de entrada será denominado, neste trabalho, **banco de dados**.

O banco de dados pode ser visto como constituído de um ou mais documentos. Um exemplo de banco de dados constituído de um único documento é o de um livro no qual o usuário está interessado em recuperar as ocorrências de uma ou mais palavras, pesquisando em todo o texto. O banco de dados pode também ser subdividido em vários documentos. Um documento, neste caso, é uma parte do banco de dados separada de outra parte por um delimitador escolhido pelo usuário. Por exemplo, pode-se colocar o delimitador de documento ao final de cada capítulo de um livro, o que faz com que cada capítulo seja visto como um documento.

Um documento, por sua vez, pode ser subdividido em campos, sendo que cada campo (ao qual pode ser dado um nome) corresponde a uma parte qualquer do documento. Um exemplo deste tipo é um banco de dados constituído de fichas padronizadas, onde cada ficha é um documento que está subdividido nos campos **autor**, **título**, **editora** e **ano de publicação**.

Algumas das principais características do sistema PATPLUS são:

1. Permite criar um índice que associa cada termo do banco de dados ao local de sua ocorrência no banco de dados.
2. Palavras comuns, que não trazem informação como "de", "a", "o" e "para", podem ser removidas do índice através do uso de um dicionário que contém tais palavras, conhecido como dicionário negativo.
3. O tamanho do índice fica entre 50% e 100% do tamanho do banco de dados do usuário.
4. Permite recuperar informação no banco de dados através de uma linguagem de consulta (apresentada na seção seguinte).
5. A operação de pesquisa de palavras no banco de dados através de casamento de padrão (pesquisa de caracteres) é natural e eficiente. Por exemplo, o sistema permite a pesquisa de qualquer frase de qualquer tamanho e, acrescentando-se ao dicionário negativo verbos de ligação, verbos que expressam solicitação e alguns outros verbos, é possível também a formulação de consultas em linguagem natural, tais como:

- . O que existe sobre *fontes energéticas alternativas*?
- . Qual é o *investimento do governo com pesquisa em agricultura*?

onde as palavras em *itálico* correspondem aos termos que efetivamente são pesquisados. Também é possível determinar as maiores seqüências de palavras que se repetem mais de uma vez no banco de dados, o que permite detectar plágio entre vários textos.

O sistema PATPLUS é constituído basicamente de três módulos:

1. um módulo de conversão de banco de dados,
2. um módulo de construção de um índice invertido em forma de

árvore Patricia (Morrison, 1968) e

3. um módulo de recuperação de informação que utiliza o índice previamente construído.

O módulo de conversão permite que os documentos subdivididos em campos sejam criados inicialmente em um formato menos mnemônico do que aquele utilizado pelo sistema e que sejam depois passados para o formato do sistema. Enquanto o usuário identifica um nome de campo através de uma expressão entre dois delimitadores de campo, o sistema identifica um nome de campo por meio de um delimitador de campo seguido de um número. Além disso, este módulo é capaz de filtrar os caracteres de controle do editor de textos WordStar (WS) (MicroPro, 1980).

Árvore Patricia é um método de pesquisa digital e sua construção no sistema PATPLUS é feita a partir de uma sequência de assinaturas, onde cada assinatura é um número inteiro que representa um termo no banco de dados textual. Pode-se converter todo o banco de dados original ou somente os termos não presentes no dicionário negativo.

O módulo de recuperação é composto por uma linguagem de consulta para a detecção das ocorrências dos termos da consulta no banco de documentos e para a manipulação dos documentos recuperados e de uma interface com o usuário que permite a elaboração da consulta e a exibição destes documentos. A etapa de detecção das ocorrências corresponde ao processo de pesquisa na árvore.

Mais detalhes sobre o sistema PATPLUS podem ser obtidos em Fusaro (1988), Ziviani (1990) e Ziviani e Gonçalves (1989).

#### **4.1.2 A Linguagem de Consulta**

A linguagem de consulta do sistema PATPLUS foi objeto da dissertação de Fusaro (1988). Os comandos definidos levam em consideração o fato de a linguagem ser destinada a um sistema de recuperação de informação de pequeno porte, cujo principal mérito está em oferecer a seus usuários grande flexibilidade quanto à estruturação de seus dados.

Os comandos da linguagem de consulta se dividem em comandos

de pesquisa e comandos de tratamento. Os comandos de pesquisa se destinam à pesquisa de um ou mais termos, correspondendo aos diversos tipos de consulta possíveis. As consultas são numeradas de forma que cada consulta é associada a um número. Os comandos de tratamento, por sua vez, permitem que o usuário examine de forma controlada o material recuperado, refine sua consulta e obtenha cópias do material selecionado como importante.

Os comandos de pesquisa são:

#### 1. Consulta simples

É a consulta compreendida de um único termo, que pode ser uma palavra, um número ou um símbolo alfanumérico. São recuperadas todas as ocorrências do termo no banco de dados. Se o banco de dados é dividido em documentos, são também obtidos os números dos documentos que possuem as ocorrências encontradas.

#### 2. Consulta composta

É compreendida de qualquer sequência de dois ou mais termos. Os termos especificados na consulta são pesquisados como um conjunto de termos adjacentes, isto é, uma única frase. São recuperadas todas as ocorrências da frase no banco de dados. Se o banco de dados é dividido em documentos, são obtidos também os números dos documentos que possuem as ocorrências encontradas.

#### 3. Consulta lógica

É formada pelos operadores lógicos E, OU e NÃO ligando dois operandos ou precedendo um operando. Os operandos podem ser consultas simples, consultas compostas, consultas lógicas, consultas em campo, consultas de proximidade, consultas truncadas e números de consultas (descritas a seguir). As operações lógicas só são aplicáveis aos bancos de dados constituídos de vários documentos. Para o operador E são recuperados os documentos que satisfazem ambos os operandos que ladeiam o operador. Para o operador OU são recuperados os documentos que satisfazem pelo menos um dos operandos que ladeiam o operador. Para os operadores NÃO ou E NÃO são recuperados os documentos que satisfazem o operando que antecede o operador e não satisfazem o operando que o segue.

#### 4. Consulta em campo

Corresponde à consulta onde o termo pesquisado deve ocorrer em um ou mais campos pré-definidos de um documento.

Limita-se, portanto, aos bancos de dados constituídos de documentos subdivididos em campos. O termo pesquisado pode ser uma consulta simples, uma consulta composta, uma consulta lógica (porém sem envolver outra consulta em campo), uma consulta truncada ou uma consulta de proximidade. São recuperados os documentos cujos conteúdos dos campos especificados na consulta satisfazem à consulta.

#### 5. Consulta truncada

É composta por um único termo cujo final é mascarado pelo usuário (máscaras iniciais e centrais não foram incluídas porque são pouco utilizadas e porque exigem implementação complexa na árvore Patricia). São recuperadas todas as ocorrências do prefixo especificado na consulta. Se o banco de dados é subdividido em documentos, são também obtidos os números dos documentos que possuem as ocorrências encontradas.

#### 6. Consulta de proximidade

É formada por dois termos ou dois conjuntos de termos ligados por um operador posicional. Um conjunto de termos é composto pelo operador lógico OU. Para o operador posicional PROX/N são recuperados os documentos que possuem, distanciados de no máximo N palavras, dois termos quaisquer ligados pelo operador. Para o operador NÃO PROX/N são recuperados os documentos nos quais nenhum dos termos que precedem o operador ocorrem a uma distância menor que N palavras dos termos que o seguem. Para o operador ANT/N a recuperação é idêntica à do operador PROX/N, porém os termos que antecedem ANT/N devem ocorrer no documento antes dos termos que o seguem.

A eficácia de um sistema de recuperação de informação é a capacidade do sistema em fornecer informações que atendam plenamente às necessidades do usuário. A eficácia é normalmente medida através dos parâmetros abrangência e precisão. A abrangência mede que proporção dos documentos relevantes presentes em uma coleção de documentos é recuperada. A precisão mede que proporção dos documentos recuperados é relevante. Por documentos relevantes entende-se aqueles documentos que estão semanticamente relacionados com a questão pesquisada.

Os comandos de pesquisa se subdividem em comandos orientados

para abrangência e comandos orientados para precisão. Os comandos orientados para abrangência são a consulta simples, a consulta lógica com operador OU e a consulta de termos truncados. Os comandos orientados para precisão são a consulta composta, a consulta lógica com operadores E e NÃO, a consulta em campo e a consulta de proximidade.

Os comandos de tratamento são:

#### 1. Apresentação do Conjunto Recuperado

Apresenta as ocorrências do(s) termo(s) pesquisado(s). O usuário pode especificar o número da consulta, os documentos pertencentes ao conjunto recuperado e os nomes de campos.

#### 2. Impressão

É análogo ao comando de apresentação, permitindo que consultas, documentos e campos sejam escolhidos. O usuário pode obter uma cópia impressa do material ou uma cópia em arquivo.

#### 3. Ordenação

Ordena os documentos da consulta especificada, de acordo com os campos indicados, em ordem ascendente ou descendente. O primeiro campo corresponde à chave primária, o segundo, à chave secundária e assim sucessivamente.

#### 4. História das Consultas

As consultas selecionadas são mostradas no vídeo ou são impressas.

#### 5. Definição de uma Sessão

Permite que uma sessão padrão seja guardada em um arquivo e seja executada posteriormente, quando necessário.

#### 6. Execução de uma Sessão

As consultas e os comandos de tratamento presentes no arquivo fornecido pelo usuário são executadas.

#### 7. Memorização de uma Sessão

Todas as consultas realizadas e suas respectivas respostas são guardadas em um arquivo. Isto permite que uma sessão seja interrompida para ser continuada depois, a partir do ponto em que foi interrompida.

#### 8. Continuação de uma Sessão

Restabelece o ambiente de uma sessão interrompida, preenchendo as estruturas do módulo de recuperação com os dados

provenientes do arquivo fornecido pelo usuário.

#### 9. Cancelamento de Sessões e Consultas

Apaga a consulta especificada e remove sua resposta das estruturas de dados do módulo de recuperação. Se uma consulta não é especificada, toda a sessão é apagada.

#### 10. Indicação de Banco de Dados

O banco de dados especificado é utilizado. O comando pode ser empregado tanto para definir o primeiro banco de dados a ser utilizado, quanto para realizar trocas posteriores.

#### 11. Apresentação do Tempo Acumulado de Consulta

Mostra o tempo decorrido desde o início da sessão até o momento da solicitação da informação.

#### 12. Auxílio

Fornece informações sobre o módulo de recuperação e sua linguagem e sobre cada argumento que pode ser adicionado.

#### 13. Finalização de uma Sessão

Finaliza uma sessão de consulta, voltando o controle ao menu principal do sistema.

Esta linguagem de consulta está parcialmente implementada sob a forma de linguagem de comandos, que é apropriada para usuários especialistas no uso do sistema, mas inacessível para usuários novatos. A descrição completa da sintaxe da linguagem, suas palavras reservadas e seus delimitadores pode ser obtida em Fusaro (1988).

## 4.2 Objetivos do Projeto

O principal objetivo deste trabalho é a elaboração de uma interface para o sistema PATPLUS específica para **usuários novatos no uso do sistema**. De acordo com a classificação de usuários apresentada no Capítulo 2, estes usuários são do tipo novato em relação ao conhecimento semântico/capacidade de uso da aplicação (Tabela 2.2). Quanto ao conhecimento sintático/capacidade de interação, o usuário deve pertencer aos perfis habilitado ou profissional, (Tabela 2.1) uma vez que, para interagir com o sistema PATPLUS via esta interface, será necessário o conhecimento dos conceitos de ações (salvar, imprimir, consultar)

e objetos (diretórios, arquivos, bancos de dados) do mundo da computação.

O usuário novato no uso do sistema PATPLUS deve ser capaz de utilizar este sistema sem nenhum treinamento e sem o auxílio de qualquer ajuda externa como, por exemplo, consulta a manuais ou a outros usuários. Para isto, a interface dispõe de um sistema de assistência em linha ao usuário, que pode ser acessado durante a interação, permitindo ao usuário resolver suas dúvidas, executar pesquisas e obter resultados.

O problema deste tipo de interface é que, após uso repetido, o usuário evolui de novato para especialista na aplicação, e pode passar a achar o sistema lento, enfadonho e restritivo. No entanto, as necessidades dos usuários novatos são prioritárias neste projeto. Outro objetivo relacionado é que a interface projetada seja tão orientada para fatores humanos quanto possível. Portanto, os aspectos abordados nos capítulos 2 e 3 desta dissertação são incorporados no projeto sempre que possível.

#### **4.3 Descrição Estrutural da Interface**

A linguagem de consulta especificada para o sistema PATPLUS, descrita na primeira seção deste capítulo, é comparável às linguagens dos sistemas de recuperação de informação mais modernos, de acordo com Fusaro (1988). Esta linguagem de consulta inclui comandos vitais a um sistema de recuperação de informação como os de consulta simples, consulta composta, consulta em campo, comando de indicação do banco de dados, comando de apresentação do conjunto recuperado, comando de impressão e comando de história, inclui também comandos que aumentam a potência e a flexibilidade do sistema como os de consulta lógica, consulta truncada, consulta de proximidade e comando de auxílio e, finalmente, comandos dedicados a usuários frequentes do sistema como os de ordenação, comandos relativos à sessão e comando de tempo. Os comandos pertencentes à interface dedicada ao usuário novato foram especificados a partir deste conjunto de comandos, tendo em vista os objetivos citados no item anterior.



A função básica de um sistema de recuperação de informação é encontrar ocorrências no banco de dados que satisfazem à condição de pesquisa do usuário e recuperar, a partir deste conjunto de ocorrências, a informação requisitada pelo usuário. Para cumprir esta função, o sistema deve fornecer aos seus usuários as seguintes operações:

1. selecionar um banco de dados
2. formular uma consulta
3. obter resultados desta consulta
4. requisitar ajuda.

Sistemas de recuperação de informação, em geral, acessam um grande número de bancos de dados e, portanto, o usuário deve estar apto a escolher de onde a informação deve ser recuperada.

Para o usuário novato no uso do sistema, o processo de formulação da consulta deve ser simples e poderoso, necessitando de um projeto elaborado. O usuário novato ficará satisfeito se ficar convencido de que a informação que obteve (ou não) é toda a que existe no banco de dados. O conjunto de comandos de consulta deve ser tal que possibilite ao usuário extrair a informação desejada do banco de dados, se ela existir. Os comandos de pesquisa devem ser em pequeno número, uma vez que, um número grande de comandos pode confundir o usuário novato. O conjunto de comandos de pesquisa apropriado para este usuário é:

1. consulta simples
2. consulta composta
3. consulta em campo
4. consulta lógica.

Quanto aos comandos de tratamento, o usuário novato quer apenas ver o que foi recuperado e ter uma cópia impressa da informação selecionada. O comando de história também é importante para que as consultas que foram feitas mas que não estão mais na tela devido à rolagem de tela, possam ser revistas. Os comandos de tratamento são:

1. comando de apresentação do conjunto recuperado
2. comando de impressão
3. comando de história.

O objetivo do projeto da interface inclui a criação de um

ambiente auto-explicativo em linha, que torne desnecessário o treinamento ou qualquer ajuda externa. Portanto, um sistema de auxílio ao usuário é parte integrante da interface. Este sistema tem dois objetivos básicos, a saber:

1. manter presente na tela, ou através de acesso fácil em linha, toda a informação necessária ao usuário durante a utilização do sistema;
2. fornecer mensagens explicativas, sensíveis ao contexto, sempre que forem requisitadas pelo usuário.

A informação necessária durante a utilização do sistema consiste de:

1. comandos disponíveis no contexto no qual o usuário está operando;
2. nome do banco de dados corrente;
3. nome dos campos dos documentos, caso existam;
4. número de documentos no banco de dados.

A informação sobre os nomes dos campos dos documentos auxilia o usuário no processo de consulta em campo, no processo de apresentação e no processo de impressão do material selecionado. O número de documentos do banco de dados dá uma dimensão do seu tamanho. O usuário deve utilizar esta informação de forma intuitiva, comparando o número de documentos que atendem à sua pesquisa com o número total de documentos no banco de dados corrente.

Dois níveis de ajuda são fornecidos na interface. No primeiro nível, são fornecidas mensagens breves que correspondem à primeira requisição de ajuda feita pelo usuário. Neste nível, o usuário pode requisitar auxílio adicional, que é fornecido pelo segundo nível de ajuda. Esta organização do sistema de ajuda em dois níveis possibilita ao usuário requisitar informação auxiliar detalhada apenas quando necessário e ter, em primeiro plano, mensagens breves e recordativas.

Todas as decisões e considerações de projeto da interface estão baseadas nas questões de fatores humanos para sistemas interativos e, em função destas questões, foram escolhidos os estilos de interação de menus e janelas para a operação do sistema. Outras técnicas como uso de *highlighting*, *blinking* e

video reverso são também utilizadas.

#### 4.4 Representação da Interface

Técnicas de especificação formal têm sido aplicadas a muitos aspectos do desenvolvimento de software. Seu valor é permitir ao projetista descrever precisamente o comportamento externo de um sistema sem especificar sua implementação interna. No entanto, tais técnicas têm sido muito pouco aplicadas na especificação de interfaces com o usuário, apesar destas interfaces serem reconhecidas como um elemento crítico em muitos sistemas de software. Projetar uma boa interface com o usuário sem uma técnica clara e precisa para sua especificação torna o projeto limitado.

Uma técnica para especificação de interfaces homem-computador deve ter as seguintes propriedades:

1. a especificação da interface com o usuário deve ser fácil de entender; em particular, deve ser mais fácil de entender e produzir do que o software que implementa a interface com o usuário;
2. a especificação deve ser precisa; não deve deixar dúvidas quanto ao comportamento do sistema para cada entrada possível;
3. deve servir a todas as fases do desenvolvimento do diálogo de forma comum e consistente;
4. a técnica de especificação deve ser potente o suficiente para expressar o comportamento de sistemas não triviais com um mínimo de complexidade;
5. deve separar o que o sistema faz (funções) de como ele faz (implementação); através desta técnica deve ser possível descrever o comportamento de uma interface com o usuário sem restringir sua implementação;
6. deve ser suportada por ferramentas interativas mas deve também ser manualmente produzível para situações nas quais as ferramentas não estão disponíveis;
7. deve ser possível construir um protótipo do sistema a partir da especificação apresentada;

8. a estrutura da especificação deve ser intimamente relacionada com o modelo mental que o usuário tem do próprio sistema.

Estudos extensivos e comparações importantes das técnicas para representação e especificação de interfaces (Jacob, 1983; Hartson & Hix, 1989) mostram que estas técnicas são baseadas, predominantemente, em um dos dois modelos formais: Backus-Naur Form (BNF) e diagramas de transição de estados. Cada um destes modelos fornece uma sintaxe para descrever fluxos de entrada dos usuários. Com o objetivo de especificar linguagens interativas, estes modelos devem ser modificados para descrever, além das entradas dos usuários, as ações do sistema e sua sequência com respeito à entrada.

A especificação da interface objeto deste trabalho será feita através de diagramas de transição de estados adaptados para representação de interfaces, conforme descrito por Jacob (1983, pp.261-264). A única modificação feita nos diagramas com relação à sua sintaxe tradicional é que cada transição é associada a uma ação: sempre que a transição ocorre o sistema desempenha a ação associada. Uma vez que o conceito de sequência é explícito em um diagrama de estados e implícito na forma BNF, o primeiro é mais adequado para especificação de diálogos porque fornece representações mais compreensíveis e que mostram de forma mais direta o sequenciamento e a estrutura da interface. Ambos os métodos são aplicáveis apenas para representação de diálogos sequenciais.

A notação empregada na presente especificação segue convenções amplamente utilizadas. Cada estado é representado por um círculo. Cada diagrama possui um estado inicial e um estado final. Os estados são rotulados dentro do círculo. Cada transição entre dois estados é mostrada com um arco dirigido, que é rotulado com o nome de um *token* de entrada, em letras maiúsculas, e, em alguns casos, uma nota contendo condições lógicas, respostas do sistema e ações. Uma dada transição de estado irá ocorrer se o *token* de entrada é recebido e a condição é satisfeita. Quando a transição ocorre o sistema exibe a resposta e desempenha a ação.

Uma transição pode ser rotulada com o nome de outro diagrama (em letras minúsculas), ao invés de um *token* de entrada. Tal transição será feita se o diagrama rotulado é percorrido com sucesso. Esta notação permite partir a especificação, possibilitando maior clareza. Na apresentação das condições para mudança de estado nomes de procedimentos em letras maiúsculas e notações simples de linguagens de programação, tais como assinalamento e comparação, podem ser utilizadas sem que as correspondentes tarefas tenham definição mais precisa. Os *tokens* de entrada podem ser definidos em uma especificação separada que captura os detalhes de baixo nível da interação usuário-computador. O *token* ENT, por exemplo, poderia ser representado pelo string "ENT", por uma tecla de função, por uma escolha em um menu exibido, etc., sem afetar a especificação mostrada.

Devido às dificuldades na edição de diagramas gráficos, a especificação será representada na forma de texto, também sugerida por Jacob (1983, pp.263). A representação textual consiste de uma lista de transições que resumem o diagrama, cada uma representada por uma linha da forma

```
s1: ENT resp: "OK" --> s2
```

assinalando uma transição do estado *s1* para o estado *s2*, com a entrada do *token* ENT e a exibição da resposta "OK". Condições e ações são especificadas de forma similar à resposta. Os nomes dos estados são escritos em letra minúscula. Ao invés de um *token* de entrada, o nome de outro diagrama pode ser dado, em letras minúsculas, significando que se aquele diagrama for transposto com êxito, a transição para o estado *s2* será feita. As características descritas acima também valem para esta notação.

O problema da representação de interfaces não está restrito às questões da linguagem. Existem aspectos visuais e perceptivos a serem representados. Os diagramas de transição de estados e a forma BNF representam relacionamentos gramaticais entre entradas do usuário como, por exemplo, sequenciamento lógico de comandos. Nenhuma das duas técnicas mostra o processo de como, por exemplo, o comando é solicitado pelo sistema e introduzido pelo usuário ou

de que forma o sistema responde com retorno semântico (mudanças na tela, no cursor ou adição de algum efeito). Para o projetista do diálogo diversos aspectos são fundamentais: navegação no sistema, restrições sintáticas e léxicas para entrada de dados, aparência dos dados exibidos na tela (gráficos, posicionamento, telas auxiliares, movimento do cursor caractere a caractere, *highlighting*, cor), conteúdo e formato das mensagens, dependência dos dispositivos e dos estilos de interação, fluxo de dados, digitação de dados, rolagem de tela, paginação, janelas, etc. Técnicas adicionais são requeridas para representar estes aspectos. Para uma representação completa da interface é necessário um conjunto de técnicas capaz de registrar o comportamento, a estrutura e os aspectos visíveis e não visíveis da interface homem-computador.

Hartson, Hix & Kraly (1990) estudaram o problema da representação física no processo de desenvolvimento do diálogo e concluíram que este é um problema não resolvido. Isto significa que, até a presente data, não existe nenhuma notação (além de código implementado) para representação completa da interface.

Estes autores consideraram então, que a estrutura do diálogo é importante em no mínimo dois domínios: o domínio comportamental e o domínio estrutural. O comportamento do diálogo é o que o usuário vê, ouve e faz durante a interação com o sistema de computador. A estrutura do diálogo determina como o sistema de computador é construído para fazer este comportamento acontecer. Desta forma, são necessárias pelo menos duas técnicas para representação completa do diálogo: uma que represente a visão comportamental e outra para representar a visão estrutural.

Hartson, Hix & Kraly (1990) sugerem que a visão comportamental seja representada por "cenários" que são formados pela imagem de cada tela, por uma identificação para cada tela e por regras que definem as entradas do usuário e as mudanças de tela baseadas nestas entradas. Diagramas derivados de máquinas de estado são utilizados para representar o aspecto estrutural.

Neste trabalho a interface será representada através de duas técnicas complementares. A primeira consiste dos diagramas de transição de estados na forma textual, que representarão a

parte da linguagem e sequenciamento. A segunda é a apresentação das telas que mostrará como as funções serão acessadas pelos usuários e demais aspectos da interface.

A interface do sistema Patplus está dividida em três módulos, a saber:

1. módulo principal
2. módulo de conversão e construção da árvore
3. módulo de recuperação.

Cada módulo será representado por um ou mais diagramas. Cada diagrama tem como função executar uma operação. Os diagramas são compostos de estados. Para cada módulo serão apresentadas as telas pertinentes.

A especificação dos *tokens* de entrada utilizados nos diagramas de transição de estados é apresentada na tabela 4.1. A segunda coluna mostra como o usuário obtém o *token* da primeira coluna.

**TABELA 4.1**

**Especificação dos *tokens* de entrada**

---

<i>TOKEN</i>	realização física
PATPLUS	digitar 'PATPLUS'
ESC	tecla <ESCAPE>
EXEC	tecla <ENTER>
F1	tecla <F1>
F5	tecla <F5>
F9	tecla <F9>
F10	tecla <F10>
PgUp	tecla <PgUp>
PgDn	tecla <PgDn>
ALFANUM	teclas de caracteres alfanuméricos
outros	opções dos menus apresentados

---

O relacionamento entre os três módulos do sistema pode ser representado na forma de um diagrama de estados que é uma abstração dos diagramas que serão apresentados para cada módulo.

#### Diagrama do Sistema PATPLUS

```

inicial : PATPLUS --> módulo 1
módulo 1: EXEC cond: se BDprepared --> módulo 3
módulo 1: EXEC cond: se NOT BDprepared --> módulo 2
módulo 2: cond: fim da preparação --> módulo 3
módulo 3: MUDA_BD/SAI --> módulo 1
módulo 1: TERMINA --> final
  
```

#### 4.4.1 Interface do Módulo Principal

O módulo principal corresponde à primeira tela da interface. O objetivo deste módulo é obter do usuário o nome do banco de dados com o qual ele trabalhará. Os dados deste módulo são resumidos na tabela 4.2 e a tela principal, correspondente a ele, é apresentada na figura 4.1.

**TABELA 4.2**

**Quadro-Resumo do Módulo Principal**

Diagrama	Função	Estados
principal	escolha do banco de dados	inicial bandados diretorio arquivos final
ajuda	fornecer informação de auxílio ao usuário	inicial ajuda1 ajuda2 final



# S I S T E M A P A T P L U S

Banco de Dados

Diretorio

C:\\$BIB\\*.\*

C:\\$BIB\\*.\*

C:\\$BIB\\*.\*

PATINTER.EXE	PPPTESTE.EXE	RETESTE.EXE	LIVROS.ERR
RETESTE.OVR	MAXMIN.EXE	RETESTE.BAK	HIPERTXT.ERR
LIVROS.ADR	LIVROS.ARR	LIVROS.CFG	LIVROS.CON
LIVROS.PAT	LIVROS.STS	LIVROS.STW	RETESTE.MAP
ARTIGOS.REL	HIPERTXT.REL	ARTIGOS.CFG	HIPERTXT.CFG

ENTER - Executa

ESC - Proxima Janela

Figura 4.1 Tela Principal

A representação textual do diagrama correspondente à tela principal é a seguinte:

Diagrama do Módulo Principal

```

inicial   : PATPLUS   ação: apresenta tela 1   --> bandados
bandados  : EXEC      cond: se EXISTE_BD(BD)   --> final
bandados  : EXEC      cond: se NOT EXISTE_BD(BD) --> bandados
bandados  : ESC                               --> diretorio
bandados  : ajuda     --> bandados
bandados  : FIM       --> inicial
diretorio: ESC        --> bandados
diretorio: EXEC      cond: se EXISTE_DIR(DIR)
                        resp: apresenta arquivos --> arquivos
    
```

```

diretorio: EXEC   cond: se NOT EXISTE_DIR(DIR)  --> diretorio
diretorio: FIM                                     --> inicial
diretorio: ajuda                                     --> diretorio
arquivos : ESC                                       --> bandados
arquivos : EXEC                                       --> final

```

#### Diagrama de Ajuda

```

inicial:          --> ajuda1
ajuda1 :  AJUDA   --> ajuda2
ajuda1 :  ESC     --> final
ajuda2 :  ESC     --> final

```

A tela 1 ou principal possui três janelas: banco de dados, diretório e arquivos. Na janela banco de dados o usuário pode especificar qualquer arquivo presente em qualquer diretório acessado pelo sistema. Na janela diretório o usuário pode fornecer o nome de qualquer diretório presente em qualquer unidade de disco acessada pelo sistema. Os arquivos deste diretório serão imediatamente listados na janela arquivos. A última linha da tela exibe os comandos executáveis no contexto corrente. O sequenciamento das ações é dado pelo diagrama acima.

Uma vez obtido o nome do arquivo que contém o banco de dados, o sistema verifica se este está ou não preparado para ser consultado. Em caso afirmativo, o sistema imediatamente apresenta ao usuário a tela principal do módulo de recuperação. Caso contrário, o banco de dados deverá sofrer o pré-processamento necessário para possibilitar a pesquisa, que consiste na execução do módulo de conversão e construção da árvore Patricia. Neste caso, a tela deste módulo será apresentada ao usuário.

#### **4.4.2 Interface do Módulo de Conversão e Construção da Arvore**

O objetivo do módulo de conversão e construção é converter o banco de dados do usuário para o formato interno do sistema Patplus e construir um índice para este banco de dados. Neste ponto o sistema já está de posse do nome do banco de dados e o

usuário deve apenas disparar o processo de conversão e construção. Este processo é denominado preparação de banco de dados e é dividido em duas etapas. Durante a sua execução, que pode ser demorada, o sistema informa ao usuário que o processo está sendo executado através de um mecanismo simples: uma barra desenhada na tela é progressivamente colorida da esquerda para a direita, à medida que o processo está sendo executado. Através do progresso da coloração na barra o usuário pode estimar o tempo necessário para o término da execução.

Além disto, o sistema tem uma função que estima o tempo necessário para a preparação do banco de dados, em função do seu tamanho. Esta função não é precisa porque não leva em conta variáveis importantes no processo como as características do ambiente e da máquina. Os dados deste módulo são resumidos na tabela 4.3 e a tela correspondente é apresentada na figura 4.2.

**TABELA 4.3**

**Quadro-Resumo do Módulo de Conversão e Construção**

---

Diagrama	Função	Estados
conversão e construção	preparação do arquivo	inicial
		final
ajuda	fornecer informação de auxílio ao usuário	inicial
		ajuda1
		ajuda2 final

---

S I S T E M A P A T P L U S

Preparacao do Banco de Dados: HIPERTXT.REL

Executando 2a. etapa...

Tamanho do Banco de Dados: 13440 bytes

Tempo de execucao estimado: 39 seg

Inicio

Fim

Figura 4.2 Tela do Módulo de Conversão e Construção

A representação textual dos diagramas do módulo de conversão e construção é a seguinte:

Diagrama Conversão e Construção

inicial: EXEC	ação: executa conversão e construção	--> final
inicial: ajuda		--> inicial
inicial: ESC	ação: volta ao módulo principal	--> final

### Diagrama Ajuda

```
inicial:          --> ajuda1
ajuda1 :  AJUDA  --> ajuda2
ajuda1 :  ESC    --> final
ajuda2 :  ESC    --> final
```

#### **4.4.3 Interface do Módulo de Recuperação**

Neste módulo o usuário pode consultar o banco de dados. Os comandos disponíveis são:

1. AJUDA - fornece texto de auxílio ao usuário
2. APAGA - apaga uma consulta digitada e não executada
3. REPETE - repete a última consulta executada
4. EXECUTA - executa uma consulta e apresenta o resultado
5. MOSTRA - mostra as ocorrências no banco de dados
6. IMPRIME - imprime trechos relevantes do banco de dados
7. HISTORIA - mostra todas as consultas realizadas
8. SAI - volta ao módulo principal do sistema
9. CAMPOS - marca os campos relevantes para o usuário

Além disso o usuário pode digitar suas consultas. Os dados deste módulo são resumidos na tabela 4.4 e sua tela principal é apresentada na figura 4.3. As telas de exibição (corresponde ao comando mostra), impressão e história são mostradas nas figuras 4.4, 4.5 e 4.6 respectivamente.

LIVROS.CON		PATPLUS 3.0						
Documentos: 1708								
Consulta		F10-Menu						
1]human computer interface		Ajuda	F1					
-----		Apaga						
OCORRENCIAS: 4	DOCUMENTOS: 4	Repete						
DOCUMENTO(S): 1238 1447 1554 1704		Executa	F5					
2]interface design		Mostra						
-----		Imprime						
OCORRENCIAS: 9	DOCUMENTOS: 9	Historia						
DOCUMENTO(S): 163 642 927 984 1314 1511 1554 1696 1704		Sai						
3]user interface management systems		Campos						
-----		<table border="1"> <tr><td>AUTOR</td></tr> <tr><td>TITULO</td></tr> <tr><td>EDITORA</td></tr> <tr><td>ANO</td></tr> <tr><td>CONTEUDO</td></tr> </table>		AUTOR	TITULO	EDITORA	ANO	CONTEUDO
AUTOR								
TITULO								
EDITORA								
ANO								
CONTEUDO								
OCORRENCIAS: 4	DOCUMENTOS: 2							
DOCUMENTO(S): 1369 1552								
4]								

Figura 4.3 Tela do Módulo de Recuperação

LIVROS.CON		PATPLUS 3.0						
Documentos: 1708								
Exibicao		F10-Menu						
AUTOR Pfaff, G. E.		Ajuda	F1					
TITULO User Interface Management Systems		Consulta						
EDITORA Springer-Verlag		Documentos						
ANO 1985		Campos						
		<table border="1"> <tr><td>AUTOR</td></tr> <tr><td>TITULO</td></tr> <tr><td>EDITORA</td></tr> <tr><td>ANO</td></tr> <tr><td>CONTEUDO</td></tr> </table>		AUTOR	TITULO	EDITORA	ANO	CONTEUDO
AUTOR								
TITULO								
EDITORA								
ANO								
CONTEUDO								
		Campo:	-					
		Documento:	1369					
		Ocorrencia:	-					
		Total Ocor:	4					
PgUp Ocor Ant		PgDn Prox Ocor						
ESC Retorna								

Figura 4.4 Tela de Exibição

TABELA 4.4

Quadro-Resumo do Módulo de Recuperação

---

Diagrama	Função	Estados
consulta	apresentação dos comandos disponíveis, elaboração e execução das consultas	inicial menu campos final
mostra	exibe ocorrências da consulta no B.D.	inicial menu consulta documentos campos final
imprime	imprime a informação definida pelo usuário	inicial menu consulta documentos campos final
história	mostra as consultas que não aparecem mais na tela	inicial menu final
ajuda	fornece informação de auxílio ao usuário	inicial ajuda1 ajuda2 final

---

A representação textual dos diagramas do módulo de recuperação é a seguinte:

### Diagrama Consulta

inicial:	ALFANUM		-->	inicial		
inicial:	F1	ação:	apresenta ajuda	-->	inicial	
inicial:	F5	ação:	executa consulta			
		resp:	mostra resultados	-->	inicial	
inicial:	F10			-->	menu	
menu	:	ajuda		-->	inicial	
menu	:	APAGA	ação:	apaga a consulta	-->	inicial
menu	:	REPETE	ação:	copia na tela a última consulta	-->	inicial
menu	:	EXECUTA	ação:	executa consulta		
		resp:	mostra resultados	-->	inicial	
menu	:	mostra		-->	inicial	
menu	:	imprime		-->	inicial	
menu	:	história		-->	inicial	
menu	:	SAI	ação:	volta ao módulo principal	-->	final
menu	:	CAMPOS		-->	campos	
menu	:	ESC		-->	inicial	
campos	:	EXEC	ação:	marca campo	-->	campos
campos	:	ESC		-->	inicial	

### Diagrama Ajuda

inicial:		-->	ajuda1	
ajuda1	:	AJUDA	-->	ajuda2
ajuda1	:	ESC	-->	final
ajuda2	:	ESC	-->	final



### Diagrama Mostra

inicial	: ESC	ação: volta ao diagrama	
		consulta	--> final
inicial	: F1	ação: apresenta ajuda	--> inicial
inicial	: F10		--> menu
inicial	: PGUP	ação: mostra a próxima	
		ocorrência	--> inicial
inicial	: PGDN	ação: mostra a ocorrência	
		anterior	--> inicial
menu	: ajuda		--> inicial
menu	: CONSULTA		--> consulta
menu	: DOCUMENTOS		--> documentos
menu	: CAMPOS		--> campos
menu	: ESC		--> inicial
consulta	: EXEC	ação: escolhe consulta	--> inicial
consulta	: ESC		--> inicial
documentos:	EXEC	ação: marca documento	--> documentos
documentos:	ESC		--> inicial
campos	: EXEC	ação: marca campo	--> campos
campos	: ESC		--> inicial

### Diagrama Imprime

inicial	: ESC	ação: volta ao diagrama	
		consulta	--> final
inicial	: F1	ação: apresenta ajuda	--> inicial
inicial	: F10		--> menu
inicial	: EXEC	ação: imprime a informação	
		selecionada	--> inicial
menu	: ajuda		--> inicial
menu	: CONSULTA		--> consulta
menu	: DOCUMENTOS		--> documentos
menu	: CAMPOS		--> campos
menu	: ESC		--> inicial
consulta	: EXEC	ação: escolhe consulta	--> inicial
consulta	: ESC		--> inicial

```
documentos: EXEC      ação: marca documento      --> documentos
documentos: ESC              --> inicial
campos      : EXEC      ação: marca campo          --> campos
campos      : ESC              --> inicial
```

Diagrama História

```
inicial: PGUP/PGDN      ação: paginação da tela  --> inicial
inicial: ESC            ação: volta ao diagrama
                        consulta          --> final
inicial: F1             ação: apresenta ajuda    --> inicial
inicial: F10           --> menu
menu      : ajuda              --> inicial
menu      : IMPRIME           ação: imprime as consultas
                        realizadas      --> inicial
```

LIVROS.CON		PATPLUS 3.0	
Documentos: 1708			
Impressao		F10-Menu	
Tecle <ENTER> para imprimir <Esc> para voltar a tela de consulta F1 para ver texto de auxilio F10 para acessar o menu		Ajuda F1 Consulta Doc os Cam 7 <div style="border: 1px solid black; padding: 5px; width: fit-content;">           AUTOR            TITULO            EDITORA            ANO            CONTEUDO         </div>	

Figura 4.5 Tela de Impressão

## Historia

## F10-Menu

## 1]HUMAN COMPUTER INTERFACE

Ajuda F1

Imprime

Ocorrencias: 4 No. de Documentos: 4  
Documentos: 1238 1447 1554 1704

## 2]INTERFACE DESIGN

Ocorrencias: 9 No. de Documentos: 9  
Documentos: 163 642 927 984 1314 1511 1554 1696 1704

## 3]USER INTERFACE MANAGEMENT SYSTEMS

Ocorrencias: 4 No. de Documentos: 2  
Documentos: 1369 1552

PgDn

Figura 4.6 Tela de História

## CAPITULO 5

### PROJETO DA INTERFACE PARA O SISTEMA PATPLUS

#### 5.1 Arquitetura de Um Sistema Interativo

A arquitetura de um sistema interativo é uma descrição de como a interface homem-computador relaciona-se com as demais partes do sistema. A lógica de um sistema interativo pode ser dividida em três componentes, conforme mostra a figura 5.1 (Hartson & Hix, 1989). O componente computacional contém a funcionalidade semântica do sistema. O componente do diálogo contém a lógica e o conteúdo do diálogo tais como telas apresentadas, mensagens e processamento da entrada. O componente de controle global governa o sequenciamento lógico entre o diálogo e a computação, em tempo de execução. Cada componente do sistema é produzido em software e mapeado em hardware.

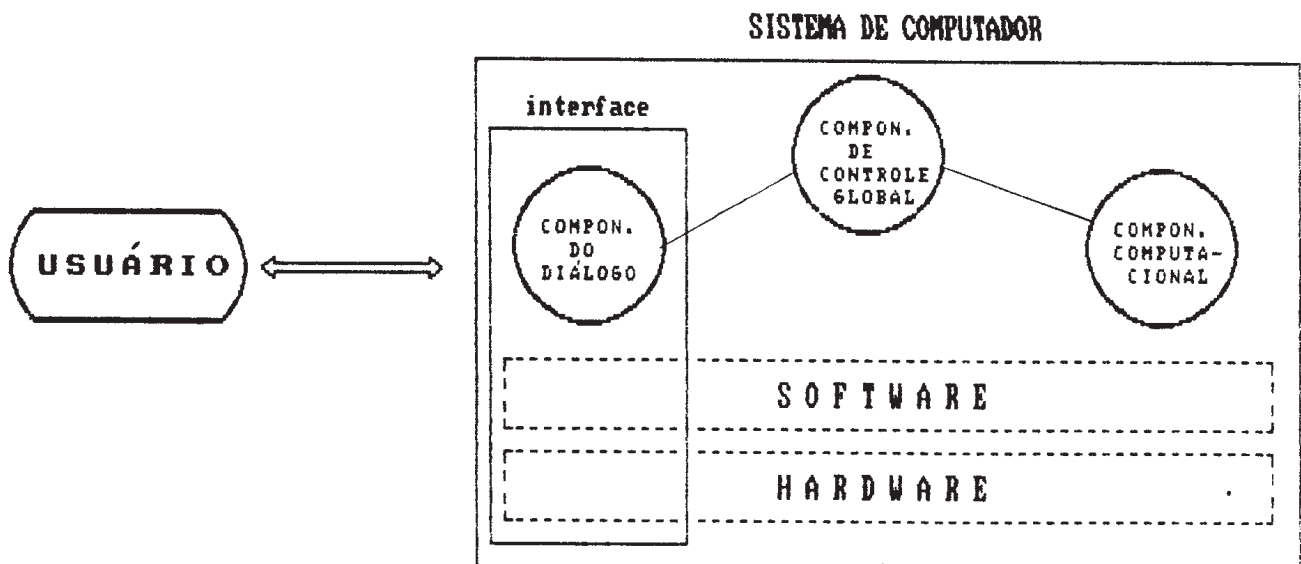


Figura 5.1 Arquitetura de um sistema interativo

O componente do diálogo é composto por uma coleção de módulos (*units*) que implementa a funcionalidade do diálogo e contém sua lógica. O componente do diálogo contém computação, mas somente computação que suporta diretamente as funções do diálogo tais como as que produzem displays, prompts, mensagens e as que extraem do usuário entradas válidas incluindo funções de aceitação, análise, validação e mapeamento das entradas. O componente computacional é composto também por uma coleção de módulos que contém a funcionalidade semântica da aplicação e não contém qualquer diálogo. O componente de controle global contém a lógica de alto nível para controlar o sequenciamento entre os módulos dos demais componentes. Na prática, ele chama separadamente procedimentos e funções dos componentes do diálogo e computacional.

A figura 5.2 mostra como a interação deve ser vista em tempo de projeto do sistema. A interação entre o usuário final e o componente do diálogo é cumprida através do que pode ser chamado de **diálogo externo** - a interface homem-computador. A separação entre componente do diálogo e componente computacional cria uma nova interface entre estes componentes e um novo tipo de diálogo através desta interface. O componente computacional, que não contém mecanismos para comunicação direta com o usuário final, empenha-se no **diálogo interno** com o componente do diálogo. Esta nova interface interna e seu diálogo especial formam a base para comunicação entre o projetista do diálogo e o projetista do programa de aplicação em tempo de projeto e são o ponto focal na ligação dos componentes computacional e de diálogo em tempo de execução.

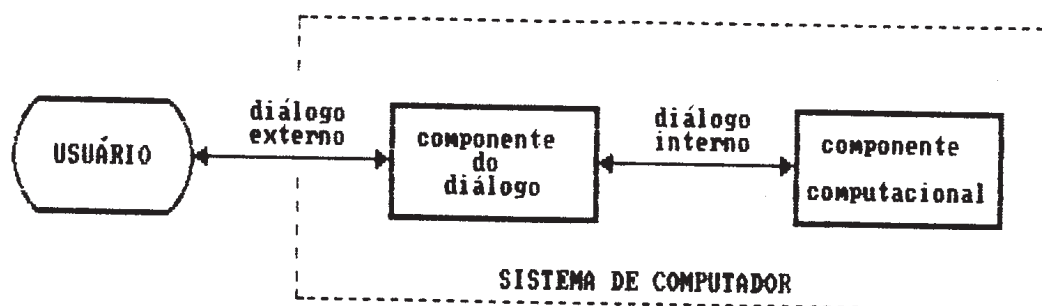


Figura 5.2 Outra visão da interação usuário-computador

É importante ressaltar que o diálogo interno não é visto em tempo de execução, mas a sua representação formal em tempo de projeto do sistema é a chave para a independência do diálogo. A comunicação entre os componentes do sistema e o usuário funciona da seguinte forma: uma vez recebido o comando do usuário, a interface traduz o comando para uma forma padrão e o envia ao programa de aplicação. A informação retornada pelo programa de aplicação em resposta às requisições do usuário é enviada à interface, que é responsável por sua exibição para o usuário. Portanto, para ambos os componentes é fundamental uma representação comum consistente do diálogo interno.

## 5.2 Modularização Interface - Sistema PATPLUS

Esta seção apresenta a modularização do sistema Patplus. Esta modularização está organizada de acordo com a proposta de uma arquitetura de um sistema interativo descrita na seção 5.1. Para tornar a descrição clara e consistente convencionaremos chamar de componente computacional os módulos de conversão (CONVERT), de construção (BUILDPAT) e de recuperação (RETRIEVE); de componente do diálogo os módulos da interface (INTERUNITS e PATINTER); e de componente de controle global os módulos CONVINTER, BUILDINTER e RETINTER e de sistema Patplus o conjunto destes módulos.

A figura 5.2 mostra os dois tipos de diálogo existentes em um sistema interativo em tempo de projeto. O diálogo externo do sistema Patplus foi descrito formalmente no capítulo 4. Esta seção descreve o diálogo interno. A estrutura do sistema Patplus, sob a visão da figura 5.1, é mostrada na figura 5.3. Os retângulos pontilhados representam, nesta figura, os módulos pertencentes ao componente computacional. Os retângulos desenhados em linha contínua representam os módulos pertencentes ao componente do diálogo. Os círculos representam os componentes de controle global.

O módulo PATINTER é o módulo principal da interface. É ele que direciona o controle do sistema para um dos componentes de controle global. A seguir, cada um dos módulos dos componentes do

diálogo e de controle global serão descritos, com ênfase para o diálogo interno. Os módulos do componente computacional estão descritos em Fusaro (1988).

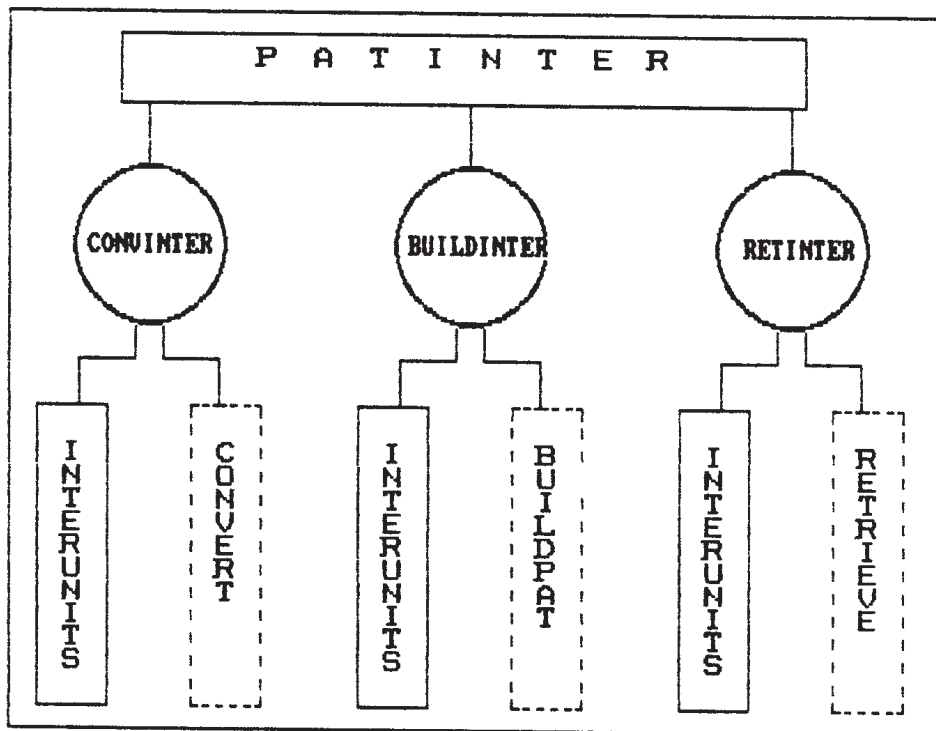


Figura 5.3 Modularização do Sistema Patplus

### 5.2.1 Componente do Diálogo

#### Módulo INTERUNITS

Este módulo é composto de *units* que fazem a comunicação sistema-usuário. São os procedimentos deste módulo que produzem as telas, as janelas, as mensagens, etc., e através deles são recebidos os comandos do usuário. As *units* e suas respectivas funções são:

- \* *DefineWindows*: esta *unit* define todas as janelas e menus do sistema.
- \* *Global*: contém declarações de constantes e tipos globais ao sistema, em especial as constantes que determinam o número das janelas e o mapa de cores.
- \* *Directory*: contém procedimentos que permitem acessar todos os

diretórios do sistema e listar todos os arquivos existentes; contém também todas as funções necessárias ao tratamento de arquivos.

- \* Interutil: constrói em tempo de execução todas as janelas e telas do sistema; também faz efeitos especiais como desaparecimento do cursor e realce de janelas.
- \* Keyboard: interpreta cada uma das teclas pressionadas pelo usuário; recebe uma tecla e devolve o caractere ASCII correspondente e seu tipo, de acordo com uma tabela; permite ao usuário digitar strings e modificá-los.
- \* Error: recebe dos procedimentos mensagens de erro e as envia ao usuário de forma padronizada.
- \* Help: implementa o sistema de ajuda ao usuário; recebe o chamamento de qualquer parte do sistema e devolve texto de auxílio sensível ao contexto no qual o usuário está operando.
- \* LêConsulta: lê uma consulta do usuário, encaminha a pesquisa, apresenta os resultados e gerencia os comandos da tela de consulta.
- \* TrataConsulta: implementa os comandos de tratamento (exibição, impressão e história).

### Módulo FATINTER

O módulo FATINTER é o módulo que controla todo o sistema. Sua função é obter do usuário o nome do banco de dados a ser pesquisado e encaminhá-lo a um dos componentes de controle global. Este módulo não faz nenhuma interação com o componente computacional, isto é, ele desconhece o componente computacional. Sua descrição na forma algorítmica é a seguinte:

```
" obtém do usuário o nome do banco de dados"  
" verifica se o banco de dados está preparado"  
" se não estiver"  
" então início"  
    " executa CONVINTER"  
    " executa BUILDINTER"  
    " fim"  
" executa RETINTER"
```



### 5.2.2 Componentes de Controle Global

Para mostrar a estrutura do sistema em termos do diálogo interno entre os módulos do componente do diálogo e do componente computacional será utilizado o modelo estrutural. As ferramentas de projeto empregadas no modelo estrutural são retângulos, ligações e soquetes. Os retângulos representam os módulos e os arquivos do sistema. As ligações representam as vias de comunicação entre módulos e arquivos. Os soquetes representam as entradas e saídas de cada módulo ou arquivo. Através das vias de comunicação, valores de parâmetros são transferidos entre módulos e arquivos. A indicação da direção da transferência é feita escrevendo-se os nomes dos parâmetros em frente aos soquetes dos dois elementos ligados: naquele que transfere, os nomes são escritos do lado de dentro do retângulo e naquele que recebe, do lado de fora. O aninhamento de módulos é feito desenhando-se um retângulo dentro do outro. Por fim, dois tipos de cruzamento de ligações são possíveis: um cruzamento onde as vias se comunicam (indicado por um ponto) e um cruzamento em que não há comunicação (indicado por uma semi-circunferência).

#### Módulo CONVINTER

Este é o componente de controle global que faz o sequenciamento entre o módulo da interface e o módulo da conversão. Ele recebe do FATINTER o nome do banco de dados. Seu diálogo interno com CONVERT está descrito na figura 5.4 e é simples: ele passa para o módulo de conversão o nome do banco de dados e recebe uma variável booleana que indica se houve erro durante a execução da conversão. Em caso afirmativo o usuário será informado sobre o erro através de um arquivo cujo nome é o mesmo nome do banco de dados e cuja extensão é 'ERR'.

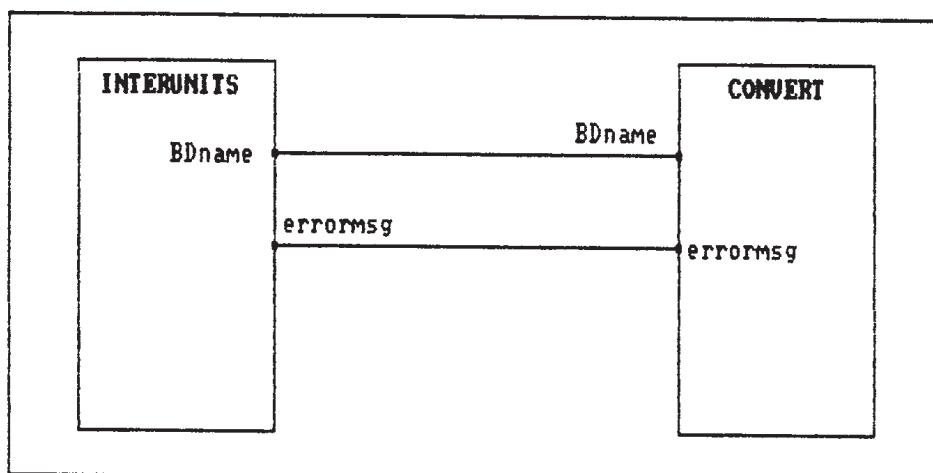


Figura 5.4 Diálogo Interno interface-CONVERT

#### Módulo BUILDINTER

Este módulo é o componente de controle global que faz o sequenciamento entre procedimentos da interface e procedimentos do BUILDPAT. Ele recebe do PATINTER o nome do banco de dados, transfere para o módulo BUILDPAT e recebe de volta duas variáveis: a primeira indica se houve ou não erro durante a execução. A segunda variável indica o tempo decorrido entre as diversas etapas do BUILDPAT e é utilizada para montar um mecanismo visual que mostra ao usuário a relação entre o tempo decorrido e o tempo que falta para o término da execução deste módulo. Este módulo é de execução muito demorada, em função do tamanho do banco de dados. O usuário precisa ser informado de que o computador está executando seu processo. Portanto, além de mostrar isso ao usuário, este mecanismo dá uma idéia do tempo que falta para que o processo termine.

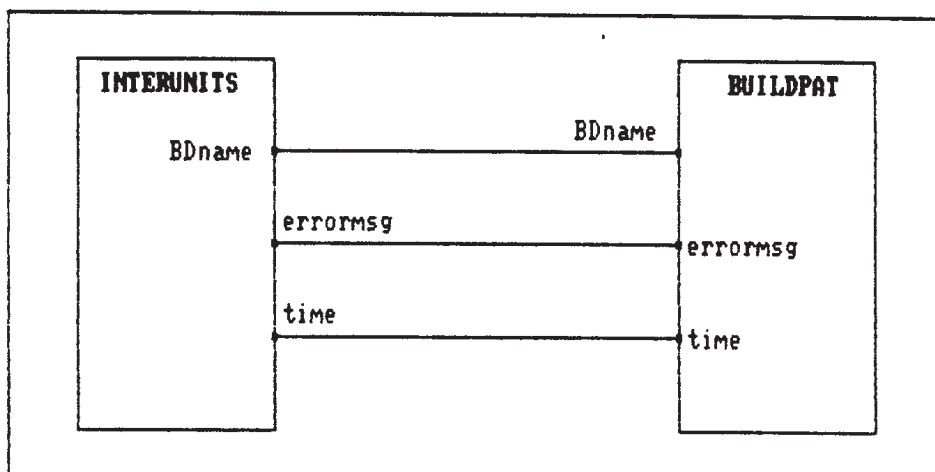


Figura 5.5 Diálogo Interno interface-BUILDPAT

#### Módulo RETINTER

O RETINTER é o componente de controle global que faz o sequenciamento entre procedimentos da interface e do RETRIEVE. Este componente de controle global é o mais complexo dos três. O diálogo interno é baseado nas estruturas de dados QUERYBUFFER, FIELDNAMEDICT, QUERYSENTENCES e QUERYLIST (descritos em Fusaro, 1988). A tabela 5.1 mostra a relação entre os comandos disponíveis para o usuário na tela de consulta e os respectivos procedimentos invocados do componente computacional. Apenas os comandos que chamam procedimentos do componente computacional foram relacionados.

O comando 'Campos' acessa a estrutura de dados FIELDNAMEDICT para obter os nomes dos campos. Os campos escolhidos pelo usuário são marcados na estrutura FIELDSET, que é consultada pelos comandos cuja especificação dos nomes de campos é relevante. As mensagens de erro são passadas do RETRIEVE para a interface e são exibidas por este componente. A figura 5.6 apresenta a estrutura do diálogo interno entre a interface e o módulo RETRIEVE.

TABELA 5.1

Relação Comandos da Interface - Procedimentos do Componente Computacional

---

Comandos	Procedimentos
Executa	SearchQuery
Mostra	ExecBrowse
Imprime	ExecPrint
História	Review

---

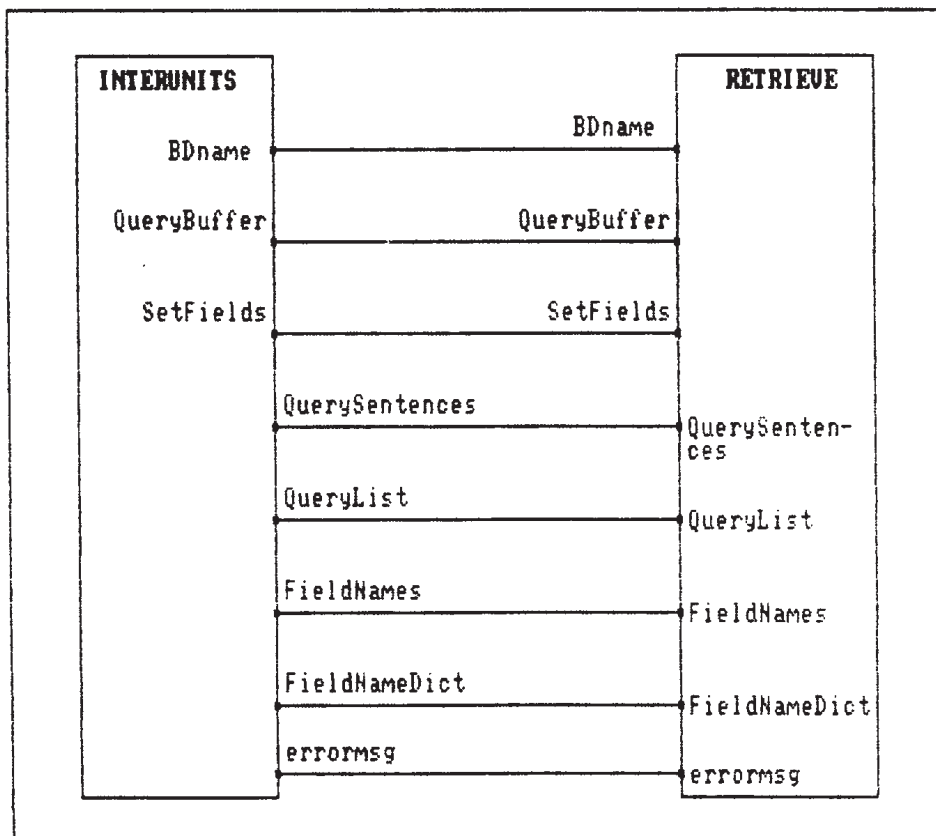


Figura 5.6 Diálogo Interno interface-RETRIEVE

### IMPLEMENTAÇÃO DA INTERFACE PARA O SISTEMA PATPLUS

Neste capítulo são discutidos alguns aspectos da implementação da interface proposta. As decisões tomadas são apresentadas e justificadas com base no aparato teórico exposto nos capítulos 2 e 3. A implementação foi feita em Turbo Pascal 5.0 (Borland, 1988) e sistema operacional DOS. Foi indispensável o uso do software WINDOW (Bigonha, 1989), composto de procedimentos para construção, manutenção e seleção de janelas e menus.

#### O banco de dados está preparado?

Esta pergunta é respondida pela função booleana `BdPrepared` pertencente à *Unit Directory*. O sistema considera que o banco de dados escolhido pelo usuário está preparado para recuperação se as seguintes condições forem simultaneamente satisfeitas:

1. se existir o arquivo correspondente ao banco de dados;
2. se existirem os seguintes arquivos: `nome.CON`, `nome.STS`, `nome.ADR`, `nome.ARR` e `nome.PAT`, onde 'nome' corresponde ao nome do arquivo que contém o banco de dados sem a extensão;
3. se a data e hora de criação do arquivo correspondente ao banco de dados for anterior ou igual à data e hora de criação do arquivo `nome.PAT`.

Se alguma destas condições não for satisfeita, o banco de dados será considerado não preparado e o sistema encaminhará o processo de preparação.

#### Orientação para o usuário novato

A orientação para o usuário novato no uso do sistema é um

objetivo deste projeto e é visível em muitos aspectos da implementação. Os mais importantes são:

- a) escolha do estilo de interação (menus em janelas);
- b) assistência ao usuário (descrita no item posterior);
- c) comandos válidos no contexto sempre presentes na tela;
- d) acesso fácil e rápido à informação de auxílio;
- e) comandos específicos para o usuário novato (secção 4.3);
- f) retorno do sistema para cada ação do usuário;
- h) o sistema toma decisões pelo usuário: o sistema decide se o banco de dados está ou não preparado para recuperação e decide também filtrar os caracteres de controle inseridos na edição do banco de dados; estas decisões pertenciam ao usuário na versão anterior.

### **Assistência ao usuário**

A assistência ao usuário é implementada de duas formas:

1. mantém presente na tela ou através de acesso fácil em linha toda a informação necessária ao usuário durante a utilização do sistema; esta informação consiste de:
  - a) comandos disponíveis no contexto no qual o usuário está operando;
  - b) nome do banco de dados corrente;
  - c) número de documentos existentes neste banco de dados;
  - d) nomes dos campos dos documentos, se houver;
2. fornece mensagens explicativas, sensíveis ao contexto, sempre que forem requisitadas pelo usuário.

A informação de auxílio ao usuário (mensagens explicativas) está implementada em dois níveis. No primeiro nível são apresentadas mensagens breves e recordativas descrevendo cada opção de entrada do usuário, bem como uma descrição geral de uso do sistema. No segundo nível é apresentada informação mais detalhada que descreve o contexto de cada comando e seu efeito, exemplo de uso e notas contendo exceções relacionadas com o uso e significado, se houver. A informação de auxílio está disponível em qualquer ponto do sistema no qual o usuário pode fornecer entrada.

A sensibilidade ao contexto é obtida tendo como parâmetro do procedimento de ajuda o endereço de onde o usuário chamou o procedimento. Este endereço consiste do número da tela. Portanto, para cada tela do sistema é apresentada uma informação sensível ao contexto daquela tela.

## **Cores**

As cores básicas escolhidas para o texto foram o amarelo e o ciano claro, para a maioria dos casos. A cor de fundo do texto é azul também na maioria das vezes. Estas combinações - amarelo sobre azul e ciano claro sobre azul - são classificadas como sendo as de melhor legibilidade de acordo com a tabela 3.3.

Para as mensagens de erro são formadas janelas sobrepostas à tela corrente. Estas janelas são construídas com fundo vermelho e texto branco. Esta combinação só é utilizada para as mensagens de erro não deixando qualquer dúvida quanto à simbolização da mensagem. As telas de ajuda são janelas empilhadas que têm fundo ciano claro e texto azul escuro. O objetivo é diferenciar estas telas e utilizar as reconhecidas propriedades tranquilizantes da cor azul.

## **Formas de entrada de dados pelo usuário**

São três as formas pelas quais o usuário entra com dados para o sistema: digitação, escolha em menus e acionamento de teclas funcionais.

A digitação se dá em duas ocasiões: a primeira, opcional, ocorre na tela principal. É facultado ao usuário digitar o nome do arquivo que contém o banco de dados ou o nome de um diretório qualquer. Se o banco de dados está no diretório corrente então não será necessária nenhuma digitação. O usuário pode escolher o arquivo desejado em um menu de arquivos exibido na tela para o diretório em questão. Se o banco de dados não estiver no diretório corrente então o usuário deverá digitar ou modificar o nome do diretório para obter o menu de arquivos ou digitar diretamente o nome do arquivo e seu endereço.

A segunda digitação é obrigatória e se refere às consultas do usuário. Uma vez que a consulta é livre e o sistema não conhece o conteúdo do banco de dados é praticamente impossível antecipar o que o usuário deseja consultar. Ele deve, obrigatoriamente, digitar sua consulta.

Para ambas as digitações o usuário dispõe de facilidades necessárias e desejáveis. Ele pode apagar um caractere com as teclas <BackSpace> ou <Del> e pode mudar o modo de inserção com a tecla <Insert> para inserir um caractere no string ou escrever um caractere sobre outro.

A escolha em menus é feita de forma usual. Um item é marcado por vez através de uma mudança de cores na marcação. Esta marca é deslocada pelo menu através das teclas de setas. Um item marcado é escolhido pela tecla <ENTER>. Para sair do menu em questão sem escolher nenhuma opção basta pressionar a tecla <Esc>. Neste sistema, dois menus permitem seleção de mais de uma opção: o menu de campos e o menu de documentos. Cada campo ou documento pode ser marcado/desmarcado pela tecla <ENTER>. Os campos e documentos escolhidos são utilizados pelo sistema nas operações de consulta, exibição e impressão. Para sair destes menus também basta pressionar <Esc>. Alguns itens de alguns menus podem ser escolhidos de forma mais direta, através de uma tecla de função. Neste caso o usuário não precisa acessar o menu para escolher o item. Basta pressionar a tecla correspondente de onde estiver. Estes itens são sinalizados colocando-se a tecla correspondente no lado direito do item no próprio menu.

As teclas utilizadas no sistema e sua função são descritas abaixo:

- <Esc> - para sair dos diversos menus e telas
- <Enter> - para escolher itens nos menus
- <setas> - para deslocar a marca de item nos menus
- <F10> - para acessar o menu de cada tela
- <F1> - para obter informação de auxílio
- <F5> - para executar uma consulta (tela de consulta)
- <PgUp>, <PgDn> - para paginar textos e menus.



## **Janelas**

O aspecto abordado no capítulo 3 quanto às janelas foi em relação à construção de janelas bonitas. Neste sistema só foram implementadas janelas bonitas para as telas que contêm informação de auxílio. As demais janelas foram projetadas tendo em vista a informação necessária na tela. Como o espaço é uma variável crítica, as janelas foram projetadas para utilizar melhor o espaço disponível, não havendo preocupação primária com a forma. No entanto, para as janelas de auxílio, não havia a preocupação com o espaço e sim com o compromisso de tornar o mais agradável possível a leitura da informação. Portanto foram utilizadas janelas bonitas nas telas de informação de auxílio.

## **Menus**

O sistema tem sete menus. O primeiro é o menu de arquivos que é formado em tempo de execução, com os arquivos presentes no diretório especificado no momento da execução. As telas de consulta, exibição, impressão e história contêm um menu cada uma. O penúltimo menu é o que apresenta os campos detectados pelo sistema nos documentos do usuário. Este menu pode ser acessado e modificado nas telas de consulta, exibição e impressão. O último menu permite ao usuário escolher os documentos que serão exibidos ou impressos para uma determinada consulta. A sequência de apresentação das opções nos menus obedece a critérios variados. Para os menus de comandos, a regra geral é que a opção de ajuda está sempre em primeiro lugar na lista de itens. Os documentos são apresentados pelo seu número, em ordem crescente. Os campos são apresentados na ordem definida pelo usuário.

## **Telas**

O sistema é composto de seis telas básicas. A organização semântica das telas é apresentada na figura 6.1.

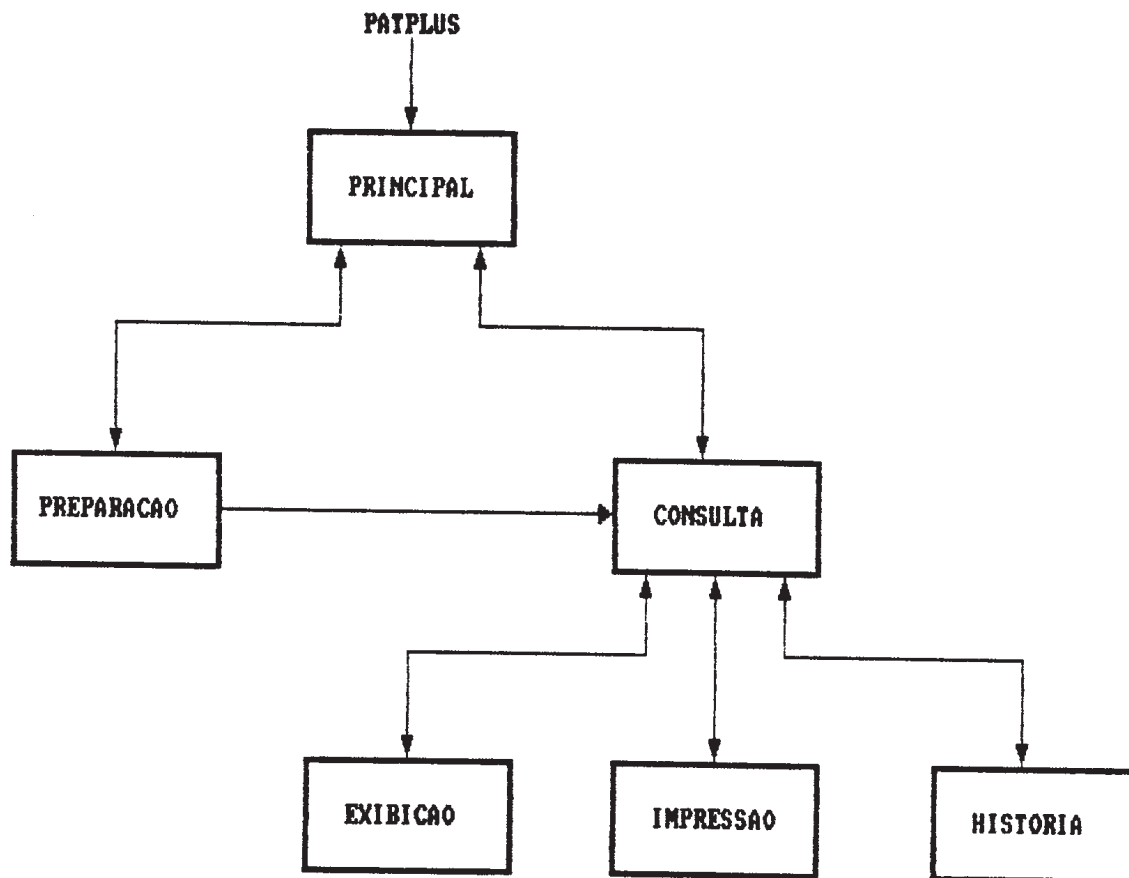


Figura 6.1 Organização semântica das telas

## Paginação

A paginação de textos em janelas e a paginação de menus é utilizada em diversas partes do sistema.

## Mensagens de Erro

Neste trabalho não foi dada nenhuma ênfase ao estudo das mensagens de erro, à descrição do erro para o usuário e opções de recuperação. As mensagens de erro provenientes do componente computacional e as mensagens de erro implementadas no componente do diálogo foram apenas padronizadas. Sempre que ocorre um erro aparece uma janela com fundo vermelho e texto branco tendo como título a palavra 'Erro'. Esta janela contém uma mensagem -

normalmente uma frase - escrita na cor branca. Esta janela permanece na tela até que o usuário pressione <Esc>. Para erros sérios, isto é, erros que impedem o sistema de progredir na sua execução, o usuário será convidado a consultar o arquivo nome.ERR, onde nome é o nome do arquivo que contém o banco de dados.

### **Retorno Semântico**

Uma característica particular deste sistema é o fato de que a preparação do banco de dados para recuperação é uma tarefa relativamente lenta e é função do tamanho do banco de dados. Esta etapa pode durar de alguns segundos a horas de execução. Por isso é necessário um retorno semântico para informar ao usuário que a operação está sendo executada. Isto é obtido através de modificações na tela. Para isto foi implementada uma barra de tempo que é colorida à medida que a execução é completada. O usuário pode facilmente extrapolar, a partir do progresso da cor na barra, o tempo necessário para finalização do processo.

### **Custo da Interface**

Uma forma de avaliar o custo da interface é através da relação entre o número de linhas do componente do diálogo e o número total de linhas de código do sistema, conforme descrito no capítulo 1. A tabela 6.1 mostra essa relação para cada módulo do sistema Patplus e para o sistema como um todo. Em média, o código da interface ocupa 47% do código do sistema. A tabela 6.2 apresenta dados sobre o tamanho do sistema em termos de Kbytes.

TABELA 6.1

Tamanho do código em número de linhas

MODULO	TAMANHO	INTERFACE
Patinter +		
Conversão	4530	54%
Construção	5820	38%
Recuperação	8990	49%
Total	19340	47%

TABELA 6.2

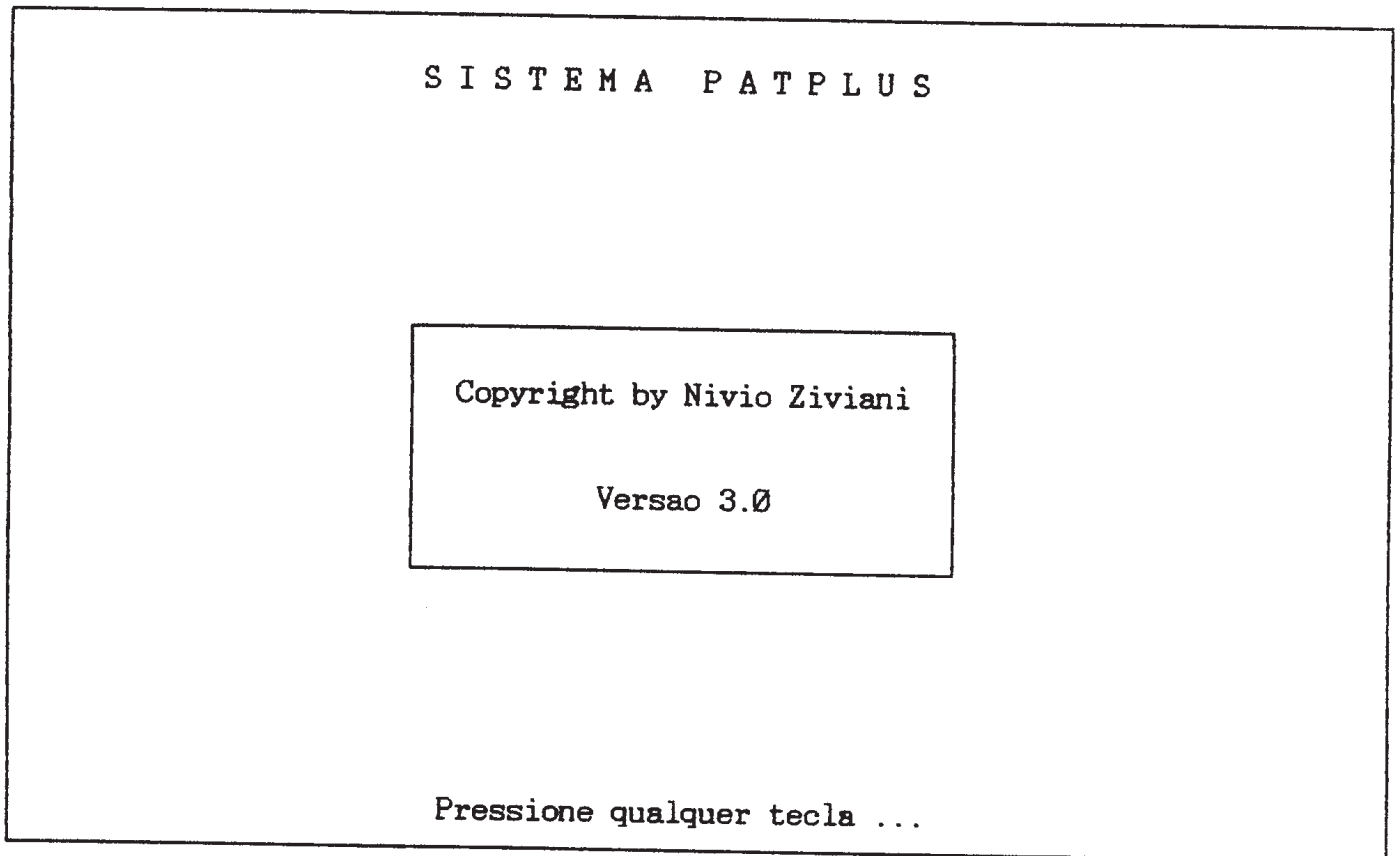
Dados do sistema Patplus em termos de Kbytes

MODULO	Código	Dados	Código Executável	Overlay	Memória Ocupada
Patinter +					
Conversão	54	32	60	--	102
Construção	44	53	48	--	113
Recuperação	66	63	72	26	173

## CAPITULO 7

### EXEMPLO DE UTILIZAÇÃO DO SISTEMA PATPLUS

Este capítulo apresenta um exemplo de utilização do sistema Patplus. Cada tela é apresentada juntamente com um texto explicativo. A primeira tela, mostrada abaixo, é a tela de apresentação do sistema. Para prosseguir basta pressionar <Enter>.



A tela seguinte à tela de apresentação é a tela principal do sistema. Nesta tela, mostrada na figura abaixo, o usuário deve informar ao sistema o nome do banco de dados com o qual deseja trabalhar. Para isto, ele pode digitar o nome do arquivo na janela Banco de Dados e pressionar <Enter>.

SISTEMA PATPLUS

\_\_\_\_\_ Banco de Dados \_\_\_\_\_

livros.con

\_\_\_\_\_ Diretorio \_\_\_\_\_

C:\\$BIB\\*.\*

\_\_\_\_\_ C:\\$BIB\\*. \* \_\_\_\_\_

F1 - Ajuda      F9 - Fim      ENTER - Executa      ESC - Proxima Janela

O usuário pode também digitar o nome de um diretório seguido de <Enter>. Neste caso, os arquivos presentes no diretório escolhido são apresentados na janela de arquivos e o usuário pode selecionar um destes arquivos. A janela Diretório apresenta como "default" o diretório corrente.

SISTEMA PATPLUS

\_\_\_\_\_ Banco de Dados \_\_\_\_\_

\_\_\_\_\_ Diretorio \_\_\_\_\_

C:\\$BIB\\*.\*

\_\_\_\_\_ C:\\$BIB\\*. \* \_\_\_\_\_

PATINTER.EXE	PPTESTE.EXE	RETESTE.EXE	LIVROS.ERR
RETESTE.OVR	MAXMIN.EXE	RETESTE.BAK	HIPERTXT.ERR
LIVROS.ADR	LIVROS.ARR	LIVROS.CFG	LIVROS.CON
LIVROS.PAT	LIVROS.SPS	LIVROS.STW	RETESTE.MAP
ARTIGOS.REL	HIPERTXT.REL	ARTIGOS.CFG	HIPERTXT.CFG

ENTER - Executa      ESC - Proxima Janela

Se o banco de dados escolhido não está preparado para pesquisa, o sistema apresenta a tela de preparação (figura abaixo). O usuário deve pressionar <Enter> para prosseguir a preparação. Para retornar à tela principal, pressionar <Esc>.

```
S I S T E M A   P A T P L U S

Preparacao do Banco de Dados: HIPERTXT.REL

Pressione <ENTER> para prosseguir

F1 - Ajuda           ENTER - Executa           ESC - Retorna
```

Após pressionar <Enter> na tela anterior, o seguinte conjunto de telas é apresentado, mostrando a sequência da preparação do banco de dados.

```
S I S T E M A   P A T P L U S

Preparacao do Banco de Dados: HIPERTXT.REL

A preparacao do banco de dados e feita em 2 etapas.
Executando a 1a. etapa...
Aguarde...
```





A etapa de recuperação é inicializada pela tela de Consulta. O banco de dados utilizado para esta demonstração é o banco de dados da biblioteca do DCC/UFMG, denominado LIVROS.CON. Na tela de consulta (figura abaixo) temos as seguintes informações relacionadas com o banco de dados: nome, número total de documentos (1708) e nome dos campos para cada documento (Autor, Título, Editora, Ano e Conteúdo).

LIVROS.CON		PATPLUS 3.0
Documentos: 1708		
Consulta		F10-Menu
1]		Ajuda F1 Apaga Repete Executa F5 Mostra Imprime Historia Sai Campos
		<div style="border: 1px solid black; padding: 5px;">           AUTOR            TITULO            EDITORA            ANO            CONTEUDO         </div>

A primeira consulta pode ser digitada. Para obter o resultado da pesquisa basta pressionar <F5>, que corresponde à escolha da opção Executa do menu. A resposta consiste do número de ocorrências do termo pesquisado no banco de dados, do número de documentos nos quais o termo ocorre e dos números destes documentos. A consulta no. 1, mostrada na figura abaixo, é um exemplo de consulta simples (com apenas um termo).

LIVROS.CON		PATPLUS 3.0
Documentos: 1708		
===== Consulta =====		===== F10-Menu =====
1]database		Ajuda F1
-----		Apaga
No. de Documentos maior do que 10		Repete
Area disponivel para 800 documentos		Executa F5
Deseja guarda-los [S/N] ? [S] s		Mostra
OCORRENCIAS: 309 DOCUMENTOS: 39		Imprime
DOCUMENTO(S): 10 39 75 109 127 155 163 164 165 169		Historia
Deseja continuar vendo [S/N]? [S] n		Sai
2]		Campos
		AUTOR TITULO EDITORA ANO CONTEUDO

As consultas 2, 3 e 4 são exemplos de consultas compostas (com mais de um termo).

LIVROS.CON Documentos: 1708	PATPLUS 3.0
Consulta	
2]human computer interface	F10-Menu
-----	Ajuda F1
OCORRENCIAS: 4 DOCUMENTOS: 4	Apaga
DOCUMENTO(S): 1171 1447 1554 1704	Repete
3]data structures	Executa F5
-----	Mostra
No. de Documentos maior do que 10	Imprime
Area disponivel para 864 documentos	Historia
Deseja guarda-los [S/N] ? [S] s	Sai
OCORRENCIAS: 86 DOCUMENTOS: 31	Campos
DOCUMENTO(S): 28 46 87 127 152 314 332 333 342 344	
Deseja continuar vendo [S/N]? [S] n	
4]interface design	
	AUTOR
	TITULO
	EDITORA
	ANO
	CONTEUDO

A consulta em campo pode ser demonstrada com o seguinte conjunto de pesquisas, todas executadas para o mesmo termo: 'KNUTH'.

- 1] Consulta simples (sem escolha de campo)
- 2] Consulta no campo Autor
- 3] Consulta no campo Título
- 4] Consulta no campo Conteúdo.

LIVROS.CON		PATPLUS 3.0
Documentos: 1708		
Consulta		F10-Menu
1]knuth		Ajuda F1
-----		Apaga
No. de Documentos maior do que 10		Repete
Area disponivel para 900 documentos		Executa F5
Deseja guarda-los [S/N] ? [S] s		Mostra
OCORRENCIAS: 17 DOCUMENTOS: 17		Imprime
DOCUMENTO(S): 282 321 438 441 442 443 466 606 758 940		Historia
Deseja continuar vendo [S/N]? [S] s		Sai
959 1323 1398 1418 1545 1553 1673		Campos
2]knuth		AUTOR
-----		TITULO
OCORRENCIAS: 9 DOCUMENTOS: 9		EDITORA
DOCUMENTO(S): 282 438 441 442 443 466 606 1323 1545		ANO
3]knuth		CONTEUDO

Pela primeira consulta sabemos que existem 17 ocorrências do termo KNUTH no banco de dados. Pela consulta no campo Autor concluímos que Knuth é o autor de 9 livros. Portanto, as 8 ocorrências restantes devem aparecer no campo Conteúdo de outros livros, que não os do Knuth, o que é confirmado pela quarta consulta.

LIVROS.CON		PATPLUS 3.0
Documentos: 1708		
===== Consulta =====		F10-Menu =====
OCORRENCIAS: 17 DOCUMENTOS: 17 DOCUMENTO(S): 282 321 438 441 442 443 466 606 758 940 Deseja continuar vendo [S/N]? [S] s 959 1323 1398 1418 1545 1553 1673		Ajuda F1 Apaga Repete Executa F5 Mostra Imprime Historia Sai Campos
2]knuth		AUTOR TITULO EDITORA ANO CONTEUDO
-----		
OCORRENCIAS: 9 DOCUMENTOS: 9 DOCUMENTO(S): 282 438 441 442 443 466 606 1323 1545		
3]knuth		
----- NENHUMA OCORRENCIA		
4]knuth		
-----		
OCORRENCIAS: 8 DOCUMENTOS: 8 DOCUMENTO(S): 321 758 940 959 1398 1418 1553 1673		
5]		

Outro exemplo de consulta em campo é o que nos informa sobre o número de livros presente no banco de dados para cada ano de edição. Para executar esta consulta basta escolher o campo Ano e digitar o ano desejado.

LIVROS.CON		PATPLUS 3.0
Documentos: 1708		
Consulta		F10-Menu
1]1990		Ajuda F1
-----		Apaga
NENHUMA OCORRENCIA		Repete
2]1989		Executa F5
-----		Mostra
OCORRENCIAS: 1	DOCUMENTOS: 1	Imprine
DOCUMENTO(S): 1626		Historia
3]1988		Sai
-----		Campos
No. de Documentos maior do que 10		AUTOR TITULO EDITORA ANO CONTEUDO
Area disponivel para 899 documentos		
Deseja guarda-los [S/N] ? [S] s		
OCORRENCIAS: 60	DOCUMENTOS: 60	
DOCUMENTO(S): 1318 1338 1359 1385 1387 1391 1392 1393 1426		
1451		
Deseja continuar vendo [S/N]? [S]		

Podemos concluir que existe neste banco de dados nenhum livro editado em 1990, apenas um livro editado em 1989, 60 livros editados em 1988, 76 livros editados em 1987, 181 livros editados em 1986 e 188 livros editados em 1985.

Para ver os documentos selecionados como resposta de cada consulta é necessário escolher a opção MOSTRA no menu da tela de consulta. A seguinte tela mostra um documento-resposta da consulta de número 1 (termo = KNUTH). Para a exibição o usuário pode escolher a consulta que deseja ver, os documentos da consulta e os campos dos documentos. Estas escolhas são feitas no menu da tela de Exibição e são opcionais. Como "default" o sistema apresenta sempre a resposta da última consulta executada, com todos os documentos e todos os campos. Neste exemplo a opção 'Documentos' foi selecionada e o sistema apresentou os documentos-resposta da referida consulta.

LIVROS.CON		PATPLUS 3.0																					
Documentos: 1708																							
Exibicao		F10-Menu																					
AUTOR Knuth, D. E. TITULO Sorting and Searching (Volume 3) - The Art of Computer Programming EDITORA Addison-Wesley Publishing Company ANO 1973 CONTEUDO Sorting Combinatorial Properties of Permutations Internal Sorting Optimum Sorting External Sorting Searching Sequential Searching Searching by Comparison of Keys Digital Searching Hashing Retrieval on Secondary Keys		Ajuda F1 Consulta Documentos Cam																					
		<table border="1"> <tr><td></td><td>282</td></tr> <tr><td>AUT</td><td>321</td></tr> <tr><td>TIT</td><td>438</td></tr> <tr><td>EDI</td><td>441</td></tr> <tr><td>ANO</td><td>442</td></tr> <tr><td>CON</td><td>443</td></tr> <tr><td></td><td>466</td></tr> <tr><td></td><td>606</td></tr> <tr><td></td><td>758</td></tr> <tr><td></td><td>940</td></tr> </table>			282	AUT	321	TIT	438	EDI	441	ANO	442	CON	443		466		606		758		940
	282																						
AUT	321																						
TIT	438																						
EDI	441																						
ANO	442																						
CON	443																						
	466																						
	606																						
	758																						
	940																						
		Campo: 1 Documento: 1545 Ocorrencia: 15 Total Ocor: 17																					
PgUp Ocor Ant		PgDn Prox Ocor																					
		ESC Retorna																					

No canto inferior direito da tela são apresentadas algumas informações sobre o documento exibido. Campo = 1 significa que o termo pesquisado ocorre no campo 1 do documento (campo Autor). O documento apresentado é o de número 1545 e esta é a 15a. ocorrência de um total de 17. Para ver a ocorrência anterior pressionar <PgUp> e, para ver a posterior, pressionar <PgDn>.

A exibição com escolha de campos é apresentada na tela abaixo. Foram escolhidos os campos Autor, Título, Editora e Ano. O documento apresentado é o de número 466.

LIVROS.CON		PATPLUS 3.0	
Documentos: 1708			
Exibicao		F10-Menu	
AUTOR Knuth, D. E. TITULO The TEXbook EDITORA Addison-Wesley Publishing Company ANO 1986		Ajuda F1 Consulta Documentos Campos	
		<div style="border: 1px solid black; padding: 5px;">           AUTOR            TITULO            EDITORA            ANO            CONTEUDO         </div>	
		Campo: - Documento: 466 Ocorrencia: - Total Ocor: 17	
PgUp Ocor Ant PgDn Prox Ocor ESC Retorna			

A tela de impressão é mostrada a seguir. Para obter esta tela é necessário escolher a opção IMPRIME do menu da tela de Consulta. A escolha de consulta, documentos e campos pode ser feita de forma idêntica à tela de Exibição. Após a escolha pressione <Enter> para obter o texto impresso.

LIVROS.CON		PATPLUS 3.0	
Documentos: 1708			
Impressao		F10-Menu	
Tecl <ENTER> para imprimir <Esc> para voltar a tela de consulta F1 para ver texto de auxilio F10 para acessar o menu		Ajuda F1 Consulta Doc os Cam 7	
		<div style="border: 1px solid black; padding: 5px;">           AUTOR            TITULO            EDITORA            ANO            CONTEUDO         </div>	



Na figura abaixo podemos observar uma tela de Consulta. Ao ser apresentado o resultado da quarta consulta, a formulação da primeira já não aparecerá na tela, devido à rolagem de tela. Para rever todas as consultas executadas basta escolher a opção HISTORIA do menu desta tela. A tela de História (seguinte) é apresentada contendo as consultas, a partir da primeira. Ao pressionar <PgDn> podemos rever as próximas consultas. Todas as consultas executadas podem ser impressas através da opção IMPRIME do menu desta tela.

LIVROS.CON		PATPLUS 3.0
Documentos: 1708		
Consulta		F10-Menu
1]human computer interface	-----	Ajuda F1
OCORRENCIAS: 4 DOCUMENTOS: 4		Apaga
DOCUMENTO(S): 1238 1447 1554 1704		Repete
2]interface design	-----	Executa F5
OCORRENCIAS: 9 DOCUMENTOS: 9		Mostra
DOCUMENTO(S): 163 642 927 984 1314 1511 1554 1696 1704		Imprime
3]user interface management systems	-----	Historia
OCORRENCIAS: 4 DOCUMENTOS: 2		Sai
DOCUMENTO(S): 1369 1552		Campos
4]man machine interfaces		
		AUTOR TITULO EDITORA ANO CONTEUDO

LIVROS.CON		PATPLUS 3.0
Documentos: 1708		
Historia		F10-Menu
1]HUMAN COMPUTER INTERFACE	-----	Ajuda F1
Ocorrencias: 4 No. de Documentos: 4		Imprime
Documentos: 1238 1447 1554 1704		
2]INTERFACE DESIGN	-----	
Ocorrencias: 9 No. de Documentos: 9		
Documentos: 163 642 927 984 1314 1511 1554 1696 1704		
3]USER INTERFACE MANAGEMENT SYSTEMS	-----	
Ocorrencias: 4 No. de Documentos: 2		
Documentos: 1369 1552		
PgDn		

A opção AJUDA é a primeira opção em todos os menus do sistema. A informação apresentada é sempre relacionada à tela ativa. A seguir são apresentadas algumas telas com texto de auxílio.

**S I S T E M A P A T P L U S**

**O SISTEMA PATPLUS**  
-----

E um sistema de recuperacao de informacao em texto completo. Qualquer arquivo constituído de caracteres ASCII e um banco de dados para o PATPLUS. O banco de dados deve ser preparado para recuperacao. A fase de preparacao e feita automaticamente pelo sistema e e executada uma unica vez para cada banco de dados, a nao ser que este sofra qualquer modificacao. Uma vez preparado, o banco de dados esta pronto para recuperacao.

Nesta tela voce deve escolher o arquivo que contem o banco de dados desejado. O sistema detecta se o banco de dados esta ou nao preparado e apresenta a tela de preparacao ou de recuperacao, conforme o caso.

Pressione F1 para obter mais informacao.

ESC - Retorna

LIVROS.CON		PATPLUS 3.0
Documentos: 1708	Consulta	F10-Menu
OCO DOC Des 959 2]knu --- OCO DOC 3]knu --- NEN 4]knu --- OCO DOC 5]	<div style="border: 1px solid black; padding: 5px;"> <p><b>Tela de Consulta</b> -----</p> <p>A recuperacao de informacao no banco de dados e feita atraves de quatro comandos:</p> <p>1.CONSULTA ao banco de dados: voce deve digitar os termos desejados e pressionar &lt;F5&gt; -executa- obtendo o numero de ocorrencias do termo no banco de dados.</p> <p>2.EXIBICAO: as ocorrencias dos termos no banco de dados sao mostradas ao usuario na tela, juntamente com os termos em torno da ocorrencia (opcao MOSTRA do menu).</p> <p>3.IMPRESSAO: as ocorrencias do termo no banco de dados sao impressas juntamente com os termos em torno da ocorrencia.</p> <p>4.HISTORIA: todas as consultas executadas ate o momento da solicitacao do comando sao mostradas na tela ou impressas.</p> <p>Pressione F1 para obter mais informacao. Esc - Retorna</p> </div>	F1  a F5  e ia  <div style="border: 1px solid black; width: 50px; height: 50px; margin: 10px auto;"></div> A DO

LIVROS.CON		PATPLUS 3.0	
Documentos: 1708			
===== Consulta =====		===== F10-Menu =====	
000			F1
DOC	Informacao de Auxilio da Tela de Consulta		
Des			
959	Para consultar o banco de dados basta digitar um ou mais	a	F5
2]knu	termos e pressionar <F5> ou escolher o item <Executa> do menu.	e	
---	A partir do menu os seguintes comandos estao disponiveis:	ia	
000	1. APAGA uma consulta digitada mas nao executada.		
DOC	2. REPETE a ultima consulta executada.		
	3. MOSTRA as ocorrencias dos termos no banco de dados.		
3]knu	4. IMPRIME as ocorrencias dos termos no banco de dados.		
---	5. HISTORIA: mostra ou imprime todas as consultas realizadas.		
NEN	6. SAI: volta a tela principal do sistema permitindo ao	A	
	usuario mudar de banco de dados ou retornar ao DOS.		
4]knu	7. CAMPOS: permite ao usuario marcar os campos para fazer	DO	
---	consulta em campos, se o banco de dados for dividido em		
000	documentos e campos.		
DOC			
5]			
		Esc - Retorna	

LIVROS.CON		PATPLUS 3.0	
Documentos: 1708			
===== Exibicao =====		===== F10-Menu =====	
AUTOR			F1
	Informacao de Auxilio da Tela de Exibicao		
TITULO	Pressione:	ta	
	<F10> para acessar o menu	ntos	
EDITOR	<F1> para obter informacao auxiliar		
	<Esc> para voltar a tela de consulta		
ANO 19	<PgUp> para mostrar a ocorrencia anterior	A	
	<PgDn> para mostrar a proxima ocorrencia	DO	
	Opcoes do Menu:		
	CONSULTA para escolher a consulta cujo resultado sera exibido		
	DOCUMENTOS para escolher os documentos que serao exibidos		
	Obs: so serao apresentados os documentos que pertencem a		
	resposta da consulta em questao.		
	CAMPOS para escolher os campos que serao mostrados		
		-	
		o: 959	
		ia: -	
		or: 8	
		Esc - Retorna	
PgUp Ocor Ant PgDn Prox Ocor ESC Retorna			

Duas mensagens de erro são mostradas nas telas seguintes. Na primeira, o usuário digitou o nome de um arquivo que não existe e pressionou <Enter>. Na segunda o usuário escolheu a opção CAMPOS na tela de Consulta, porém o banco de dados não está dividido em documentos e campos.

S I S T E M A   P A T P L U S

Banco de Dados

crisrina.doc

ERRO

C: Arquivo nao encontrado. Pressione ESC.

C:\\$BIB\\*.\*

F1 - Ajuda      F9 - Fim      ENTER - Executa      ESC - Proxima Janela

CAP4.DOC	PATPLUS 3.0
Documentos: 1	
Consulta	F10-Menu
<p>OCORRENCIAS: 64    DOCUMENTOS: 1</p> <p>DOCUMENTO(S): 1</p> <p>2]sistema patplus</p> <p>OCORRENCIAS: 14</p> <p>DOCUMENTO(S): 1</p> <p>3]interface</p> <p>OCORRENCIAS: 30    DOCUMENTOS: 1</p> <p>DOCUMENTO(S): 1</p> <p>4]projeto</p> <p>OCORRENCIAS: 7    DOCUMENTOS: 1</p> <p>DOCUMENTO(S): 1</p> <p>5]</p>	<p>Ajuda F1</p> <p>Apaga</p> <p>Repete</p> <p>Executa F5</p> <p>Mostra</p> <p>Imprime</p> <p>Historia</p> <p>Sai</p> <p>Campos</p>
<p>Nao ha divisao em campos.</p> <p>Pressione ESC</p>	

## CAPITULO 8

### CONCLUSÃO

Neste trabalho foi apresentada uma exposição sobre interfaces homem-computador e uma interface para usuários novatos do sistema Patplus. Esta interface tem como objetivo básico permitir que um usuário novato do sistema utilize-o de forma satisfatória sem o uso de qualquer ajuda externa como, por exemplo, consulta a manuais ou a outros usuários. Para alcançar este objetivo vários aspectos do desenvolvimento de interfaces foram estudados e implementados.

A principal questão em oposição à produção de uma interface para usuários novatos é a sua inadequação para usuários que já conhecem o sistema em alguma proporção (usuários gerenciais, administrativos e especialistas). Para analisar esta questão devemos considerar as características de uso do sistema Patplus. O sistema Patplus não é um sistema de uso diário contínuo para a grande maioria dos seus usuários. Este é um sistema com característica de uso intermitente. A única exceção que deve ser feita é para o caso do sistema ser utilizado como um sistema de recuperação de informação comercial e por pesquisadores profissionais que atuam como intermediários entre o sistema e o usuário. Neste caso a interface deveria ser completamente diferente.

Para a maioria dos casos, na qual o usuário final da informação obtida é o usuário do sistema, isto é, o próprio usuário pesquisa a informação desejada, pode-se dizer que esta interface é mais adequada mesmo para usuários não novatos do que a interface em linguagem de comandos. Considerando o usuário como intermitente, esta interface tem as vantagens de ser mais recordativa do que a anterior, mais agradável e com mais recursos para socorrê-lo caso sua memória falhe.

Portanto, esta interface pode ser perfeitamente utilizada para os demais usuários, guardando ressalva à exceção feita.

desde que:

1. sejam incorporados os demais comandos especificados para o sistema por Fusaro (1988) e que atendem a usuários não novatos;
2. o sistema tenha uma alta taxa de resposta tornando-o, neste aspecto, competitivo com a interface em linguagem de comandos;

A resposta do sistema para uma seleção do usuário deve ser rápida. Para isto, existem implicações a nível de hardware e a nível de software. A nível de hardware pode-se obter terminais de vídeo com velocidades de transmissão cada vez maiores, o que não é difícil, devido ao extraordinariamente rápido desenvolvimento da tecnologia de vídeo. A nível de implementação o requerimento de resposta rápida tem fortes implicações: quando uma janela é recuperada e mostrada, não há tempo de compor dinamicamente a tela obtendo dados de vários pontos do banco de dados.

Para trabalhos futuros fica como sugestão a generalização da interface para todo o universo de usuários.

## REFERENCIAS BIBLIOGRAFICAS

- ACM CHI'83** (1983). "Conference on Human Factors in Computing Systems". Boston, Mass., December. ACM, New York.
- ACM CHI'85** (1985). "Conference on Human Factors in Computing Systems". San Francisco, Calif., April. ACM, New York.
- ACM CHI'86** (1986). "Conference on Human Factors in Computing Systems". Boston, Mass., April. ACM, New York.
- ACM CHI'87 + GI** (1987). "Conference on Human Factors in Computing Systems". Toronto, Ontario, Canada, April. ACM, New York.
- ACM CHI'88** (1988). "Conference on Human Factors in Computing Systems". Washington, D.C., May. ACM, New York.
- ACM Computing Surveys** (1981). Special Issue: The Psychology of Human-Computer Interaction, 13(1), March.
- ACM Computing Surveys** (1989). 21(1), March.
- ACM SIGGRAPH** (1986). "Workshop on Software Tools for User Interface Management". Seattle, Wash., November. ACM, New York. In **Computer Graphics** 21, 2 (Apr.), pp.71-147.
- ACM SIGGRAPH** (1988). "Symposium on User Interface Software". Banff, Alberta, Canada, October. ACM, New York.

- BAECKER, R. (1980). "Towards a Characterization of Graphical Interaction", In **Readings in Human-Computer Interaction - A Multidisciplinary Approach**, Baecker, R. M. & Buxton, W.A.S., Eds., Morgan Kaufmann Publishers, Inc., 1987.
- BAECKER, R.M. & BUXTON, W.A.S. (1987). **Readings in Human-Computer Interaction - A Multidisciplinary Approach**. Morgan Kaufmann Publishers, Inc., California.
- BIGONHA, R.S. (1989). **WINDOWi 1.2 - Janelas e Menus em Turbo Pascal**, Relatório Técnico RT028/89, Departamento de Ciência da Computação, UFMG.
- BORLAND International (1988). **Turbo Pascal Reference Guide - Version 5.0**.
- BRITTS, S. (1987). "Dialog Management in Interactive Systems: A Comparative Survey", **ACM SIGCHI Bulletin**, 18(3), pp.30-42.
- CARD, S.K., PAVEL, M., & FARRELL, J.E. (1985). "Window-based Computer Dialogues". In **Readings in Human-Computer Interaction - A Multidisciplinary Approach**, Baecker, R. M. & Buxton, W.A.S., Eds., Morgan Kaufmann Publishers, Inc., 1987.
- CAREY, T.(1982). "User Differences in Interface Design", **IEEE Computer**, November, pp.49-58.
- CHI, ULI H. (1985). "Formal Specification of User Interfaces: A Comparison and Evaluation of Four Axiomatic Approaches", **IEEE Transactions on Software Engineering**, 11(8), pp.671-685.
- CHIGNELL, M.H., HANCOCK, P.A., & LOEWENTHAL, A. (1989). "An Introduction to Intelligent Interfaces". In **Intelligent Interfaces - Theory, Research and Design**, Hancock, P.A. & Chignell, M.H.(Eds.), North-Holland.



- Communications of the ACM** (1983). Special Issue: "Working Toward Successful Human-Computer Interface", 26(4), April.
- COUTAZ, J. (1985). "Abstractions for User Interface Design", **IEEE Computer**, September, pp.21-34.
- CROUCHER, C. (1986). "Problems of Subject Access: (ii) user studies and interface design", **Program**, 20(2), pp.211-224.
- DRAPER, S.W. & NORMAN, D.A. (1985). "Software Engineering for User Interfaces", **IEEE Transactions on Software Engineering**, 11(3), March, pp.252-258.
- EBERTS, R.E. & EBERTS, C.G. (1989). "Four Approaches to Human Computer Interaction". In Hancock, P.A. & Chignell, M.H.(Eds.), **Intelligent Interfaces - Theory, Research and Design**, North-Holland.
- ELKERTON, J. & WILLIGES, R.C. (1989). "Dialogue Design for Intelligent Interfaces". In **Intelligent Interfaces - Theory, Research and Design**, Hancock, P.A. & Chignell, M.H., Eds., North-Holland.
- FENCHEL, R. (1980). "Integral Help for Interactive Systems", **Report UCLA-ENG-8051, Computer Science Department, University of California**, September, Los Angeles.
- FOLEY, J.D. (1987). "Models and Tools for the Designers of User-Computer Interfaces", **Report GWU-IIST-87-03, Department of EE & CS, George Washington University**, Washington, D.C., March.
- FOLEY, J.D. & van DAM, A. (1982). **Fundamentals of Interactive Computer Graphics**, Addison-Wesley, Reading, Mass.
- FOLEY, J.D. & WALLACE, V.L. (1974). "The Art of Natural Graphic Man-Machine Conversation", **Proceedings of the IEEE**, 63(4), pp. 462-471.

- FUSARO, P.S. (1988). "Uma Linguagem de Consulta para Um Sistema de Recuperação de Informação em Texto Completo". Dissertação de Mestrado, Departamento de Ciência da Computação da UFMG.
- GAIT, J. (1986). "Pretty Pane Tiling of Pretty Windows", **IEEE Computer**, September, pp.9-14.
- GREEN, M. (1986). "A Survey of Three Dialog Models", **ACM Transactions on Graphics** 5(3), July, pp.244-275.
- GREENBERG, S. & WITTEN, I. H. (1988). "Directing the User Interface: How People Use Command-Based Computer Systems", in **Man-Machine Systems - Analysis, Design and Evaluation**, Preprints of the IFAC/IFIP/IEA/IFORS Conference, Oulu, Finland, June, Volume II, pp.299-305.
- GOULD, J.D. & LEWIS, C. (1985). "Designing for Usability: Key Principles and What Designers Think", **Communications of the ACM**, 28(3), March, pp.300-311.
- HARTSON, H.R. (1989). "Control and Communication in User Interface Management", **IEEE Software**, 6(Jan.), pp.62-70.
- HARTSON, H.R. & HIX, D. (1989). "Human-Computer Interface Development: Concepts and Systems for Its Management", **ACM Computing Surveys**, 21(1), March, pp.5-92.
- HARTSON, H.R., HIX, D. & KRALY, T. M. (1990). "Developing Human-Computer Interface Models and Representation Techniques", **Software - Practice and Experience**, 20(5), May, pp.425-457.
- HCI Hawaii (1984). "First International Conference on Human-Computer Interaction". Honolulu, Hawaii, August. International Commission on Human Aspects in Computing.

- HCI Hawaii** (1987). "Second International Conference on Human-Computer Interaction". Honolulu, Hawaii, August. International Commission on Human Aspects in Computing.
- IBM Systems Journal**, (1981). Special Issue: "Human Factors", 20(2).
- IEEE Computer**, (1982). Special Issue: "Human-Computer Interaction", 15(11), November.
- IEEE Software**, (1989). Special Issue: "Development Human-Computer Interfaces", 6(January).
- INTERACT'84** (1984). "First IFIP Conference on Human-Computer Interaction. London, September. International Federation for Information Processing.
- INTERACT'87** (1987). "Second IFIP Conference on Human-Computer Interaction. Stuttgart, September. International Federation for Information Processing.
- JACOB, R.J.K. (1983). "Using Formal Specifications in Design of a Human-Computer Interface", **Communications of the ACM**, 26(4), April, pp.259-264.
- JARKE, M. & VASSILIOU, Y. (1985). "A Framework for Choosing a Database Query Language", **Computing Surveys**, 17(3), September.
- MAEDA, K., MIYAKE, Y., NIEVERGELT, J. & SAITO, Y. (1984). "A Comparative Study of Man-Machine Interfaces in Interactive Systems", **ACM SIGCHI Bulletin**, 16(2), pp.44-61.
- MCCRACKEN, D.L. & AKSCYN, R.M. (1984). "Experience with ZOG Human-Computer Interface System", **International Journal of Man-Machine Studies**, 21, pp.293-310.
- MICROPRO International Corporation (1980). **WordStar User's Guide**.

- MONK, A. (1984). **Fundamentals of Human-Computer Interaction**. Academic Press, Florida.
- MORRISON, D. R. (1968). "PATRICIA - Practical Algorithm To Retrieve Information Coded In Alphanumeric", **Journal of the ACM** 15(4), pp. 514-534.
- MYERS, G. J. (1978). **Composite/Structured Design**. Van Nostrand Reinhold Company, New York.
- NEWMAN, W. M. & SPROULL, R.F. (1979). **Principles of Interactive Computer Graphics**, 2nd ed., McGraw-Hill, New York.
- NORMAN, D.A. (1983). "Design Principles for Human-Computer Interfaces". In **Readings in Human-Computer Interaction - A Multidisciplinary Approach**, Baecker, R.M. & Buxton, W.A.S., Eds., Morgan Kaufmann Publishers, Inc., 1987, pp.492-501.
- NORMAN, D. A. (1987). "Some Observations on Mental Models". In **Readings in Human-Computer Interaction - A Multidisciplinary Approach**, Baecker, R.M. & Buxton, W.A.S., Eds., Morgan Kaufmann Publishers, Inc., 1987, pp. 241-244.
- NORMAN, K.L., WELDON, L.J. & SHNEIDERMAN, B. (1986). "Cognitive Layouts of Windows and Multiple Screens for User Interfaces", **Int. J. Man-Machine Studies**, 25, pp.229-248.
- PERLMAN, G. (1989). "Evaluating How Your User Interfaces Are Used", **IEEE Software**, January, pp.112-113.
- PPAFF, G., ED. (1985). **User Interface Management Systems**, Springer-Verlag, Berlin.
- RICH, E. (1984). "Natural-Language Interfaces", **IEEE Computer**, September, pp.39-47.

- RUBINSTEIN, R. & HERSH, H. (1984). "Design Philosophy", Chapter 2 of **The Human Factor: Designing Computer Systems for People**, Burling, MA: Digital Press, pp.12-20. In **Readings in Human-Computer Interaction - A Multidisciplinary Approach**, Baecker, R. M. & Buxton, W.A.S., Eds., Morgan Kaufmann Publishers, Inc., 1987.
- SCHNEIDER, M. L. (1984). "Ergonomic Considerations in the Design of Command Languages", In **Human Factors and Interactive Computer Systems**, Yannis Vassiliou, Ed., Ablex Publishing Corporation, New Jersey.
- SHNEIDERMAN, B. (1984). "The Future of Interactive Systems and the Emergence of Direct Manipulation", In **Human Factors and Interactive Computer Systems**, Yannis Vassiliou, Ed., Ablex Publishing Corporation, New Jersey.
- SHNEIDERMAN, B. (1987). **Designing the User Interface - Strategies for Effective Human-Computer Interaction**. Addison-Wesley, Reading, Massachusetts.
- SMITH, D.C., IRBY, C., KIMBALL, R. & VERPLANK, B. (1982). "Designing the Star User Interface", **Byte**, April, pp.242-282.
- TESLER, L. (1981). "The Smalltalk Environment", **Byte**, August, pp.90-147.
- VASSILIOU, Y. (1984). **Human Factors and Interactive Computers Systems**. Ablex Publishing Corporation, New Jersey.
- VASSILIOU, Y. & JARKE, M. (1984). "Query languages - a taxonomy", In **Human Factors and Interactive Computer Systems**, Yannis Vassiliou, Ed., Ablex Publishing Corporation, New Jersey.
- WILLIAMS, G. (1983). "The Lisa Computer System", **Byte**, February, pp.33-50.

- WILLIAMS, G. (1984). "The Apple Macintosh Computer", **Byte**, February, pp.30-54.
- ZIVIANI, N. (1990). "Um sistema para recuperação eficiente de informação em textos", **Revista Brasileira de Computação**, Rio de Janeiro, 5(3), pp.3-15.
- ZIVIANI, N. & GONÇALVES, C.C. (1989). "Paginação de Árvores Binárias", IX Congresso da Sociedade Brasileira de Computação, Uberlândia, M.G., pp. 367-384.
- ZWICKER, R. & REINHARD, N. (1990). "Interfaces Inteligentes: perspectivas para novas formas de aprendizado e uso de sistemas", **Revista Brasileira de Computação**, Rio de Janeiro, 5(3), pp.17-25.