

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Programa de Pós-Graduação em Engenharia de Produção

WALTER DE ABREU CYBIS

**A IDENTIFICAÇÃO DOS OBJETOS
DE INTERFACES HOMEM-COMPUTADOR
E DE SEUS ATRIBUTOS ERGONOMICOS**

**Tese submetida à Universidade Federal de Santa Catarina para a
obtenção do Grau de Doutor em Engenharia de Produção**



0.227.280-1

UFSC-BU

Florianópolis, SC

Maio de 1994

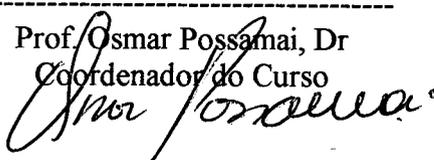
A Identificação dos Objetos das Interfaces Homem-Computador e de seus Atributos Ergonômicos

Walter de Abreu Cybis

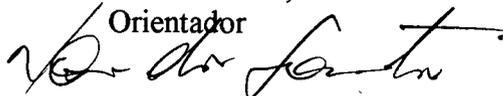
Esta tese foi julgada adequada para a obtenção do Título de
Doutor em Engenharia de Produção
e aprovada em sua forma final pelo Programa de Pós-Graduação

Banca Examinadora

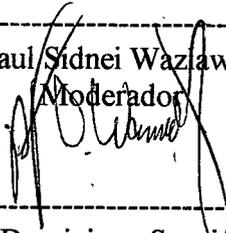
Prof. Osmar Possamai, Dr
Coordenador do Curso



Prof. Neri dos Santos, Dr.
Orientador



Prof. Raul Sidnei Wazlawick, Dr
Moderador



Prof. Dominique Scapin, Dr.
Examinador Externo



Prof. Daltro Nunes, Dr
Examinador Externo



Prof. Leila Amaral Gontijo, Dr.



para Rose e Pedro

Agradecimentos

Este trabalho de tese representa a primeira iniciativa concreta do acordo de cooperação técnica, envolvendo as equipes de ergonomia do Departamento de Engenharia de Produção e Sistemas da Universidade Federal de Santa Catarina e do Institut National de Recherche en Informatique et en Automatique de Paris, França.

O rumo principal nesta atividade foi acertado em comum acordo entre este autor, o Prof. Neri dos Santos, no Brasil e M. Dominique Scapin, na França.

Naquele país, a manutenção do projeto deveu-se ao apoio incondicional de Mme. Suzanne Sebillotte, Mlle Willemina Visser, Mme Françoise Détienne e M. Jean Marc Robert. No mesmo barco estavam os companheiros de todas as horas Christian Bastien, Françoise Darses, Nathalie Bonnardel, Latifa Talleb, Hamid Hammouche e tantos outros. A assistência logística ficou por conta de Mme. Marie-Pierre Laborne e Mme. Marie Françoise Loubressac.

Os ecos do Brasil vinham da Prof. Leila Gontijo e do colega Ricardo Monteiro, sempre de plantão no correio eletrônico.

Espalhados pelo mundo estavam meus irmãos, meus pais e familiares

Correndo o risco do esquecimento quero agradecer a todos pelas diversas formas de apoio recebidos durante estes anos de tese.

RESUMO

Esta monografia apresenta um modelo de objetos de interação abstratos ergonômicos -OIAe-, componentes lógicos de uma arquitetura para a geração automática de interfaces homem-computador ergonômicas. Eles favorecem a aplicação do raciocínio baseado em regras ergonômicas no processo de concepção e de avaliação destas interfaces, pois são identificados no universo abstrato das recomendações ergonômicas. Assim, a correspondência entre estes objetos e seus atributos com as regras ergonômicas torna-se efetiva. Ao contrário dos objetos de interação convencionais, que estão voltados para a lógica de funcionamento dos sistemas, os OIAe representam componentes voltados para a lógica de utilização das interfaces com o usuário. Em consequência, eles têm sua concepção teórica baseada no entendimento de um sistema interativo como uma ferramenta simbólica, cujos sinais são manipulados pelos usuários para modificar o estado de outros sinais. Sinais computacionais, os OIAe são implementados como agentes interativos PAC, cujas dimensões são renomeadas; Conteúdo, Controle e Expressão. A tipologia proposta organiza as classes de sinais deste modelo segundo os enfoques funcional, estrutural e formal. Como funções, eles modelam os componentes de uma ferramenta genérica, sendo assim definidas as classes de mostradores, controles, comandos e painéis de controle. Eles podem ser estruturas elementares ou compostas, que apresentam dois tipos de formas; as básicas e as auxiliares. Esta monografia se encerra com a apresentação de HyperAttribut, um aplicativo destinado a apoiar as ações de projeto e de avaliação de interfaces homem-computador com base no modelo de OIAe.

ABSTRACT

This work introduces the notion of Ergonomic Abstract Interaction Objects - AIO^e - the logical components of a system architecture concerning the automatic human-computer interface (HCI) conception. They propose a view of the HCI to support the application of ergonomic reasoning to their conception and evaluation. In opposition to conventional interaction objects, that concerns the functioning of interactive systems, the ergonomic abstract interaction objects supports the utilisation aspects of such systems. Identified on the abstract universe of ergonomic recommendations, they result from the understanding of an interactive system as a symbolic tool. Computational signs, the AIO^e are turned operational as PAC agents, whose dimensions are here renamed as: Content, Control and Expression. The proposed typology organises these objects as tool components, that propose structure and form aspects. Thus, they are elementary or composite objects that present basic and auxiliary forms and belong to classes of functions: display, controls, commands and dialogue panels. The last part of this work is dedicated to the description of HyperAttribut, a AIO^e model development and maintenance tool. It also supports the object guided human-computer interfaces evaluation and specification tasks.

SUMÁRIO

| | |
|---|-----------|
| APRESENTAÇÃO..... | 1 |
| PRIMEIRA PARTE - ERGONOMIA E INTERFACES HOMEM-COMPUTADOR.... | 5 |
| 1 - ERGONOMIA - OS CONHECIMENTOS DISPONÍVEIS..... | 6 |
| 1.1 - As Teorias Cognitivas..... | 6 |
| 1.1.1 - A percepção..... | 7 |
| 1.1.2 - As estruturas mnêmicas..... | 8 |
| 1.1.3 - O aprendizado..... | 9 |
| 1.1.4 - As formas de raciocínio..... | 10 |
| 1.1.5 - O controle da atividade..... | 11 |
| 1.2 - Os Modelos Aplicados..... | 12 |
| 1.2.1 - O modelo do processador humano..... | 12 |
| 1.2.2 - A teoria da ação..... | 14 |
| 1.2.3 - O modelo do conhecimento semântico e sintático..... | 15 |
| 1.2.4 - O modelo de linguagem..... | 16 |
| 1.2.5 - O modelo de N. Sisson..... | 17 |
| 1.2.6 - Os modelos - GOMS e Keystroke -..... | 18 |
| 1.4 - Conclusões..... | 19 |
| 2 - ERGONOMIA - AS FORMAS DE INTERVENÇÃO..... | 20 |
| 2.1 - A Abordagem Ergonômica..... | 20 |
| 2.1.1 - Os Critérios Ergonômicos..... | 20 |
| 2.1.1.1 - Condução..... | 22 |
| 2.1.1.2 - Carga de trabalho..... | 22 |
| 2.1.1.3 - Controle..... | 23 |
| 2.1.1.4 - Adaptabilidade..... | 23 |
| 2.1.1.5 - Gestão de erros..... | 23 |
| 2.1.1.6 - Homogeneidade/Consistência..... | 24 |
| 2.1.1.7 - Significância dos Códigos..... | 24 |
| 2.1.1.8 - Compatibilidade..... | 24 |
| 2.1.2 - As Recomendações Ergonômicas..... | 24 |
| 2.2 - Um sistema de conformidade ergonômica..... | 26 |
| 2.3 - A proposta da arquitetura -Archie-..... | 28 |
| 2.3.1 - O modelo da tarefa..... | 29 |
| 2.3.2 - A base de conhecimento em ergonomia..... | 30 |
| 2.3.3 - Os métodos e estratégias de projeto e de avaliação..... | 31 |
| 2.3.4 - O modelo de objetos de interface..... | 32 |

| | |
|--|-----------|
| 2.4 - Conclusões | 33 |
| 3 - IHC - PERSPECTIVA COMPUTACIONAL..... | 35 |
| 3.1 - Os componentes funcionais..... | 35 |
| 3.1.1 - Núcleo Funcional..... | 36 |
| 3.1.2 - Controle do diálogo..... | 37 |
| 3.1.3 - Objetos de interação..... | 39 |
| 3.1.4 - Sistema de janelas..... | 41 |
| 3.2 - Modelos de arquiteturas para os componentes funcionais..... | 43 |
| 3.2.1 - Modelo de máquinas abstratas..... | 44 |
| 3.2.2 - Os modelos de linguagem..... | 45 |
| 3.2.3 - O modelo Multi-agentes..... | 46 |
| 3.2.3.1 - O modelo P.A.C..... | 47 |
| 3.3 - As ferramentas para a construção de interfaces..... | 50 |
| 3.3.1 - Toolboxes..... | 51 |
| 3.3.2 - Esqueletos de aplicação..... | 52 |
| 3.3.3 - Geradores de interface..... | 53 |
| 3.4 - Conclusões | 56 |
| 4 - IHC - PERSPECTIVA SEMIÓTICA..... | 58 |
| 4.1 - A semiótica..... | 58 |
| 4.1.1 - Sinais e esquemas semióticos..... | 59 |
| 4.2 - A semiótica computacional..... | 65 |
| 4.2.1 - Sinais Computacionais..... | 67 |
| 4.2.2 - Os sinais computacionais elementares..... | 68 |
| 4.2.2.1 - Sinais interativos..... | 68 |
| 4.2.2.2 - Sinais atores..... | 68 |
| 4.2.2.3 - Sinais controles..... | 69 |
| 4.2.2.4 - Sinais objetos..... | 69 |
| 4.2.2.5 - Sinais figurativos e fantasmas..... | 69 |
| 4.2.3 - Os sinais computacionais compostos..... | 69 |
| 4.2.3.1 - Os concorrentes - cena, vista e foco..... | 70 |
| 4.2.3.2 - Os sequenciais - tarefa, ação e indicador..... | 71 |
| 4.2.4 - O caos e o diálogo..... | 72 |
| 4.3 - Conclusões | 72 |
| SEGUNDA PARTE - O MODELO DE OIA^E..... | 74 |
| 5 - DESENVOLVENDO O MODELO DE OIA^E..... | 75 |
| 5.1 - Objetivos e definição..... | 75 |

| | |
|--|------------|
| 5.2 - Materiais e métodos..... | 76 |
| 5.2.1 - Etapa de definições estruturais..... | 76 |
| 5.2.1.1 - A morfologia de um OIA ^e | 77 |
| 5.2.1.2 - A tipologia de OIA ^e | 78 |
| 5.3 - A identificação dos OIA^e | 81 |
| 5.4 - Resumo do capítulo | 81 |
| 6 - A PERSPECTIVA ESTRUTURAL..... | 83 |
| 6.1 - A classe Estrutura..... | 83 |
| 6.2- A classe Elemento | 85 |
| 6.3 - A classe Composição..... | 85 |
| 6.4.1 - A classe Gerência..... | 86 |
| 6.4.2 - A classe Correlação | 87 |
| 6.6 - O resumo do capítulo | 87 |
| 7 - A PERSPECTIVA FORMAL..... | 89 |
| 7.1 - As classes de formas básicas..... | 89 |
| 7.1.1 - As formas sonoras | 90 |
| 7.1.1.1 - A Locução | 90 |
| 7.1.1.2 - O Motivo Melódico | 91 |
| 7.1.1.3 - O Efeito Sonoro..... | 92 |
| 7.1.2 - As formas visuais..... | 92 |
| 7.1.2.1 - As formas gráficas elementares: o ícone | 92 |
| 7.1.2.2 - As formas tipográficas elementares: as palavras..... | 95 |
| 7.1.2.3 - As formas gráficas elaboradas: | 95 |
| 7.1.2.3.1 - O Texto | 96 |
| 7.1.2.3.2 - A Lista | 96 |
| 7.1.2.3.3 - A Tabela | 97 |
| 7.1.2.3.4 - O Gráfico elaborado..... | 98 |
| 7.1.2.3.5 - O Desenho/Diagrama | 99 |
| 7.1.2.3.6 - O Diagrama de fluxo | 99 |
| 7.1.2.3.7 - O Mapa..... | 100 |
| 7.2 - As Formas Auxiliares | 102 |
| 7.2.1 - A Cor | 102 |
| 7.2.2 - O Estilo | 104 |
| 7.2.3 - A Textura | 104 |
| 7.2.4 - O vídeo reverso | 105 |
| 7.2.5 - A Intermitência | 105 |
| 7.3 - Resumo do capítulo | 105 |

| | |
|--|------------|
| 8 - A PERSPECTIVA FUNCIONAL..... | 107 |
| 8.1 - Os Mostradores | 107 |
| 8.1.1 - O Mostrador de Informação..... | 108 |
| 8.1.1.1 - O Rótulo..... | 109 |
| 8.1.1.1 - A Mensagem..... | 110 |
| 8.1.2 - O Mostrador de Dados | 111 |
| 8.1.2.1 - O Mostrador de Dados Individual | 111 |
| 8.1.2.2 - O Mostrador de grupos de dados..... | 112 |
| 8.1.2.3 - O Radar, o termômetro e o alerta..... | 113 |
| 8.2 - Os Controles | 114 |
| 8.2.1 - O controle manipulável - o cursor | 115 |
| 8.2.2 - Os controles editáveis | 116 |
| 8.2.2.1 - O campo de dados..... | 116 |
| 8.2.2.1.1 - Os campos condutor e inspetor..... | 117 |
| 8.2.2.2 - O Campo de Texto..... | 118 |
| 8.2.2.4 - O Campo Gráfico..... | 119 |
| 8.2.3 - Os controles selecionáveis..... | 119 |
| 8.2.3.1 - O botão de seleção..... | 119 |
| 8.2.3.2 - O item de seleção..... | 120 |
| 8.2.3.3 - O grupo de Seletores..... | 121 |
| 8.2.3.4 - A lista de seleção | 122 |
| 8.3 - Os Comandos..... | 123 |
| 8.3.1 - Os comandos editáveis: linha e área de comando | 124 |
| 8.3.2 - Os comandos selecionáveis | 124 |
| 8.3.2.1 - O botão de comando | 124 |
| 8.3.2.2 - A opção de menu | 125 |
| 8.3.2.3 - O grupo de comandos | 126 |
| 8.3.2.4 - O painel de menu | 127 |
| 8.3.2.4.1 - A página de menu..... | 128 |
| 8.3.2.4.2 - A barra de menu..... | 129 |
| 8.4 - Os Painéis de Controle | 130 |
| 8.4.1 - A Janela..... | 130 |
| 8.4.1.1 - A janela de aplicativo | 131 |
| 8.4.2 - A caixa de diálogo | 132 |
| 8.4.2.1 - As caixas de mensagem | 133 |
| 8.4.2.2 - A caixa de Ação/Tarefa..... | 134 |
| 8.4.2.3 - A caixa de Formulário..... | 134 |
| 8.5 - O resumo do capítulo | 135 |
| 9 - DISCUSSÕES | 137 |
| 9.1 - O resumo das propostas apresentadas..... | 137 |
| 9.1.1 - OIA ^e como facilitador do raciocínio ergonômico | 137 |
| 9.1.2 - OIA ^e como recurso para a usabilidade | 139 |

| | |
|---|------------|
| 9.1.3 - OIA ^e como sinal computacional..... | 140 |
| 9.2 - Pesquisas futuras..... | 141 |
| 9.2.1 - A validação do modelo de OIA ^e | 141 |
| 9.2.2 - O desenvolvimento do aplicativo HyperAttribut..... | 142 |
| 9.2.3 - A ampliação e o refinamento da base de recomendações ergonômicas..... | 144 |
| 9.2.4 - A ligação utilização/funcionamento..... | 145 |
| 9.2.5 - Em busca de uma nova dimensão de OIA ^e ; os sequenciais..... | 145 |
| 10 - Referências Bibliográficas..... | 146 |

A IDENTIFICAÇÃO DOS OBJETOS DE INTERFACES HOMEM-COMPUTADOR E DE SEUS ATRIBUTOS ERGONÔMICOS

Apresentação

O presente trabalho de tese busca trazer as contribuições da ergonomia cognitiva às soluções dos problemas de projeto de interfaces homem-computador¹. Este objetivo justifica-se pelas deficiências que as IHC apresentam sob o ponto de vista ergonômico e pelo importante papel que estas interfaces assumem para a produtividade e a saúde no trabalho informatizado.

A interface homem-computador é determinante das estratégias e do desempenho do usuário em sua tarefa. A implantação de sistemas ou de produtos de informação deficientes sob o ponto de vista de utilizabilidade pode ser relacionada entre as causas da baixa produtividade e do fraco retorno que caracterizam muitos dos investimentos na informatização em geral.

Scapin (Scapin, 1986) identifica as seguintes causas para as deficiências das IHC:

- *conhecimento incompleto sobre a tarefa e sobre os usuários do trabalho a ser informatizado.*
- *ausência de métodos e ferramentas lógicas específicas para a concepção e avaliação de interfaces com o usuário.*
- *concepção segundo uma orientação funcional em detrimento da operacional.*

A ergonomia² mostra-se habilitada a enfrentar estas causas pois integra dados e conhecimentos sobre o homem no trabalho com o objetivo de conceber produtos e ferramentas que proporcionem um balanço ótimo entre conforto, segurança e eficiência.

Historicamente esta disciplina tem tido participação no desenvolvimento de sistemas informatizados como os de controle de centrais nucleares e do tráfego aéreo. Nestes, o caráter

¹ Uma interface homem-computador engloba todos os aspectos dos sistemas informatizados que influenciam a participação do usuário em suas tarefas (Scapin, 1986) Nela estão incluídos os dispositivos de hardware, os programas de entradas e saídas, a arquitetura do diálogo, os manuais, os cursos de treinamento e o suporte.

² segundo definição de Wisner (Laville, 1977)

crítico exige erros mínimos e eficiência máxima. Atualmente o processo de informatização da sociedade se acelera. Em consequência, pessoas sem um treinamento específico passam a ter necessidade de acesso à dispositivos informatizados existentes no trabalho, em casa, em locais públicos, etc. Assim a ergonomia, que até há pouco tempo só poderia ser considerada no desenvolvimento de sistemas de grande porte, passa a ter seu lugar no tempo de processamento e no espaço de memória das estações de trabalho atuais.

Apesar das possibilidades de ordem técnica e do consenso sobre a importância da qualidade ergonômica, esta disciplina continua afastada das decisões do projeto. Como visto neste trabalho, as possibilidades de alteração deste quadro dependem do exame e do esclarecimento sobre as formas de intervenção ergonômica na concepção de sistemas interativos e de um entendimento maior sobre a natureza das interfaces homem-computador.

O sucesso de uma intervenção ergonômica no projeto destas interfaces depende em boa parte do tipo do apoio disponível. As recomendações ergonômicas revelam-se como a forma de conhecimento mais aplicada e por isso mesmo a mais utilizada por ergonomistas e projetistas de informática. Elas correspondem à pequenos textos ou frases curtas (guidelines) que resultam de aplicação de conhecimentos teóricos às questões de projeto e avaliação de interfaces. Seu formato é especialmente favorável à definição de regras de produção.

Em termos gerais, a proposta deste trabalho refere-se à especificação de um ambiente de apoio para a geração de interfaces homem-computador ergonômicas. A arquitetura deste ambiente incorpora uma base de regras de produção derivadas das recomendações ergonômicas. A proposta específica refere-se à definição de um conjunto de objetos de interação adequado ao raciocínio ergonômico, entendido como o disparo das regras ergonômicas.

A primeira questão que se coloca frente ao objetivo específico diz respeito à portabilidade das configurações realizadas por um sistema de geração de interfaces face às diferentes plataformas de programação existentes. Cada plataforma propõe objetos de interação distintos.

Uma alternativa de solução refere-se à noção de objeto de interação abstrato -OIA-. Esta entidade resulta da interseção entre todos os conjuntos de objetos de interação (concretos) disponíveis nas plataformas de desenvolvimento atuais (variações de Windows, X-

Windows e MacApp). Teoricamente as especificações em termos de OIA podem ser traduzidas nos termos dos recursos oferecidos por uma plataforma de programação específica.

Entretanto, sob o ponto de vista do raciocínio ergonômico, verifica-se que a correspondência entre os OIA, seus atributos e as recomendações ergonômicas não é absolutamente efetiva. De fato, as entidades referidas nestas recomendações fazem parte do modelo mental de um ergonomista, não de um ambiente de programação.

O efetivo aproveitamento das recomendações ergonômicas como base de regras de um ambiente de geração de interfaces depende da definição de componentes lógicos que se aproximem dos imaginados pelos ergonomistas.

Os objetos de interação abstratos ergonômicos - OIA^e - representam a proposta inédita deste trabalho. Eles definem um conjunto de objetos de interação abstratos particularmente adequados para o raciocínio ergonômico sobre a concepção de interfaces homem-computador. Este tipo de raciocínio é facilitado pois se estabelece uma correspondência efetiva entre os OIA^e e as recomendações ergonômicas que estão em suas origens.

Os atributos de um OIA^e não são os mesmos de um objeto de interação concreto ou abstrato. Suas naturezas são diferentes. Enquanto estes últimos estão voltados para as questões relativas ao funcionamento do sistema, os OIA^e estão voltados para a sua utilização. Na utilização, as questões básicas se referem ao significado e às relações entre sinais.

A Semiótica traz os elementos teóricos para a construção de um OIA^e. Segundo esta perspectiva, um sistema interativo é visto como um tipo especial de sistema simbólico. Seus sinais são manipuláveis pelo usuário e modificáveis pela ação de um outro sinal (Andersen, 1990).

Um sinal é caracterizado por uma formação triangular; planos de conteúdo e de expressão, correlacionados na mente do interpretador. A semelhança de um sinal, um OIA^e apresenta planos de conteúdo e de expressão. No entanto estes dois planos estão explicitados no computador, onde são mantidos coerentes através de um terceiro plano de controle. O mérito desta morfologia reside exatamente na explicitação dos conteúdos pretendidos pelos projetistas, que passam a fazer parte dos atributos configuráveis de uma interface homem-computador

Um OIA^e é construído segundo a noção de agente PAC - (Coutaz, 1990), e representa uma entidade tridimensional de planos de conteúdo, de expressão e de controle. Sua taxionomia é igualmente tridimensional, com hierarquias de classes construídas à partir do enfoque na função, na estrutura e na forma de um OIA^e. Os exemplares de classes e de atributos identificados nas recomendações ergonômicas representam os recursos que o modelo de OIA^e propõe para a configuração da usabilidade de uma interface homem-computador.

Na perspectiva estrutural os OIA^e se dividem em classes de elementos e de composições. Estas são definidas a partir das partes resultantes de suas divisões. Da divisão de uma composição resultam elementos. Da divisão de um elemento resultam partes. As partes nunca ocorrem isoladamente.

A perspectiva formal propõe classes de formas básicas e de formas auxiliares. A escolha de uma forma auxiliar é determinada pela forma básica predefinida.

Os componentes da perspectiva funcional são diferenciados conforme sua capacidade de interação com o usuário e com o núcleo funcional. Eles modelam os componentes de uma ferramenta genérica através de classes de mostradores, controles, comandos e de painéis de controle.

A trajetória desta monografia se encerra com a proposta da adoção dos OIA^e como primitivas de uma possível programação orientada a sinais. Este estilo de programação, proposto originalmente por Andersen em seu livro "A Theory of Computer Semiotics: semiotic approaches to construction and assessment of computer systems" representa uma especialização do paradigma de programação orientada a objetos. A programação orientada a sinais se justifica quando se verifica que um sistema interativo está repleto de sinais (que representam objetos). O elo de ligação entre os sinais e as primitivas trazidas pelas recomendações ergonômicas (os OIA^e) reside no enfoque da lógica de utilização.

Primeira Parte - Ergonomia e Interfaces Homem-Computador

A primeira parte desta monografia apresenta a disciplina ergonômica e o seu objeto de estudo particular; as interfaces homem-computador.

Os dois primeiros capítulos analisam os tipos de conhecimento disponíveis e as formas de intervenção ergonômica na concepção de interfaces homem-computador. A proposta de um ambiente computacional para a geração de interfaces ergonômicas, é identificada como forma de intervenção econômica e compatível com a cultura dos projetistas de informática. O aspecto inédito desta proposta fica por conta dos objetos de interação abstratos ergonômicos -OIA^e, os componentes lógicos do ambiente de geração que efetivamente favorecem o raciocínio baseado nas regras ergonômicas.

Os capítulos III e IV apresentam uma interface homem-computador a partir das perspectivas computacional e semiótica. Esta apresentação tem por objetivo esclarecer sobre aspectos técnicos e teóricos envolvidos com a construção dos OIA^e.

1 - ERGONOMIA - OS CONHECIMENTOS DISPONÍVEIS

Este capítulo apresenta um apanhado geral dos conhecimentos teóricos sobre os processos cognitivos que podem ser aplicados nas intervenções ergonômicas para a concepção/avaliação de interfaces homem-computador.

Assim, são apresentadas as contribuições trazidas pelas ciências cognitivas, classificadas segundo as que fornecem um quadro teórico e as que propõem modelos aplicados. As primeiras buscam formalizar as estruturas e os tratamentos que caracterizam os sistema cognitivo humano. Os Modelos aplicados são desenvolvidos com o objetivo de auxiliar os projetistas na compreensão dos processos e necessidades cognitivas durante a interação com o computador.

1.1 - As Teorias Cognitivas

Assim como os conhecimentos sobre a fisiologia da mão e do braço são importantes no projeto de uma ferramenta manual, também os conhecimentos sobre as características humanas no tratamento da informação são importantes no projeto de um software interativo. Considerar o usuário significa conhecer além das informações provenientes da análise ergonômica do trabalho (idade, sexo, formação específica, conhecimentos, etc...), também àquelas ligadas às suas habilidades e capacidades em termos cognitivos. Na medida em que se pretende o computador como uma extensão do cérebro humano, é fundamental conhecer o papel que estas habilidades e capacidades desempenham na realização da tarefa informatizada.

Nos últimos anos a psicologia interessou-se pelo estudo do tratamento da informação. A procura das leis gerais sobre o comportamento (comportamentalismo) é complementada, não sem controvérsias, pela procura dos mecanismos que explicam o seu funcionamento (cognitivismo).

Os próximos parágrafos apresentam um resumo das teorias sobre o funcionamento cognitivo humano, baseado nos volumes I e II do *Traité de Psychologie Cognitive* (Richard, Bonnet, & Ghiglione, 1990).

A noção de representação caracteriza a abordagem cognitivista. As representações são formadas a partir de estímulos perceptivos tratados em uma sequência de processos cognitivos denominados ascendentes ou sensoriais. A representação pode ser definida como a

interpretação da realidade, sobre a qual se baseiam tratamentos descendentes ou simbólicos, que visam a produção de novos conhecimentos e a realização de ações.

1.1.1 - A percepção

O organismo toma conhecimento do mundo através do tratamento da informação sensorial. A elaboração do conhecimento perceptivo demarca a noção de sistema perceptivo. Assim a percepção abrange o conjunto de estruturas e tratamentos pelos quais o organismo impõe um significado aos dados sensoriais. O sistema perceptivo é caracterizado por um conjunto de sistemas autônomos, cada um especializado no tratamento de um tipo de estímulo.

O tratamento da informação visual é organizado segundo os níveis: neuro-sensorial, perceptivo e cognitivo. O nível neuro-sensorial envolve a transformação dos traços elementares da estimulação visual em primitivas visuais que são estruturadas seguindo diversos mecanismos. Os Gestaltistas propõem leis que descrevem as condições de aparecimento de grupamentos. Estas incluem a proximidade, a similaridade, a continuidade e a conectividade. A percepção de contornos, a segregação figura-fundo e a ocorrência de ilusões ótico-geométricas são também fenômenos da estruturação pré-semântica. Mesmo que possam corresponder à aparência de um objeto, elas ainda não permitem sua identificação. Para tanto são necessários uma representação espacial (3D) e os conhecimentos prévios sobre a função do objeto. Só aí o indivíduo tem acesso à representação fonológica e lexical sobre sua denominação.

O sistema auditivo humano recebe as informações de fontes sonoras simultâneas de maneira seletiva. As representações acusticamente coerentes, denominados objetos ou "imagens" auditivas, são organizados em processos paralelos e sequenciais. Nos processos paralelos este sistema organiza os eventos sonoros segundo sua amplitude, frequência, forma espectral e posição. Os processos sequenciais lidam com sucessões de eventos acústicos percebidos na forma de um fluxo. Os componentes de um fluxo sonoro apresentam continuidade, como em uma melodia, e são determinados por relações de frequência, cadência, intensidade, conteúdos espectrais, etc.

A percepção da linguagem falada está organizada na forma de uma série de sucessivos processos de codificação. A nível neuro-sensorial ocorre a codificação neuronal dos estímulos fonéticos. A informação sobre a estrutura espectral destes índices é extraída e estocada numa memória sensorial de curto termo. Isto permite a análise dos índices acústicos pertinentes que são confrontados com os traços fonéticos característicos de uma linguagem específica. Ocorre

então a filtragem das variações fonéticas que não são características, de maneira a isolar as unidades silábicas. A nível lexical se dão os tratamentos de acesso ao léxico e de identificação das palavras. A nível sintático ocorre a integração das informações lexicais e sintáticas com a interpretação da mensagem recebida.

As representações elaboradas pelo sistema perceptivo são tratadas em processos descendentes que visam compreender uma situação ou decidir sobre uma ação. O sistema cognitivo apresenta estruturas mnêmicas e formas de tratamento variadas.

1.1.2 - As estruturas mnêmicas

Duas classes de modelos procuram descrever a natureza e o formato das unidades mnêmicas e suas regras de armazenamento e de recuperação da informação. O modelo cibernético/computacional também chamado de modelo de Von Neumann, descreve a memória humana à semelhança da memória de um computador. O modelo biônico/conexionista - explica a memória a partir da neurofisiologia do cérebro humano, com neurônios e sinapses. O funcionamento do modelo connexionista é determinado pelas relações de entrada-saída das unidades de tratamento e pelos pesos das conexões. Estas redes são capazes de modificarem sua própria conectividade em função de uma situação externa ou de sua atividade interna. Elas tendem assim a assumir novos estados em função das restrições de uma situação específica. Este modelo de memória propõe um modo de armazenamento onde a informação é distribuída sobre um conjunto de ligações sinápticas.

O modelo cibernético propõe a existência de três tipos de memória: os registros sensoriais (RS), as memórias de curto termo (MCT) e a memória de longo termo (MLT). Esta proposição tem resistido aos diferentes movimentos da psicologia cognitiva durante as últimas décadas. Em sua versão original a informação que é liberada pelo sistema perceptivo, é estocada em um registro sensorial de capacidade limitada. A parte que é selecionada para um tratamento mais elaborado é estocada na memória de curto termo. Daí, a informação pode ainda ser estocada em um registro permanente, concebido como uma rede de conceitos. Eles representam a base de conhecimentos do indivíduo.

A capacidade da MCT é de 6 a 7 itens e seu esquecimento ocorre em poucos segundos. Esta declaração define a MCT como um registro de evocabilidade indiferente ao formato da informação e passivo ao nível de recuperação requerido. Já a memória de trabalho - MT - ou memória intermediária é definida como um centro de tratamentos. Esta é composta de dois

subsistemas escravos especializados, um nos tratamentos verbais e outro nos tratamentos visual-espacial. Um executor central é capaz de manter certas informações em um alto nível de evocabilidade.

A permanência da informação na memória de longo termo - MLT - não está sujeita a limitações de ordem temporal. Isto não implica em uma acessibilidade permanente. A acessibilidade depende das condições em que ocorreu o registro da informação tanto no nível episódico como no semântico. A memória episódica guarda o conhecimento de ordem procedural essencialmente dinâmico e automatizado. O efeito do contexto (intrínseco, interativo, psicológico) é o fator determinante da recuperação da informação na memória episódica. Um bom desempenho depende da compatibilidade entre as condições contextuais no momento do registro e no momento da recuperação. A memória semântica estoca conhecimentos declarativos organizados segundo uma rede de proposições conceituais. O acesso à informação independe do contexto e se dá pela ativação de um de seus nós e pela propagação desta ativação aos nós vizinhos. O esquecimento em memória permanente deve-se a um processo de revisão mental, ao aumento do número e da semelhança das representações e pela incompatibilidade entre os contextos de codificação e de recuperação.

Os tratamentos descendentes que caracterizam os processos cognitivos visam a produção de novos conhecimentos e a realização de ações.

1.1.3 - O aprendizado

O aprendizado é definido como o processo de modificação tanto do conhecimento (declarativo) como do comportamento (procedural) do organismo. Ele resulta das interações com o ambiente. A nível de conhecimentos o aprendizado define a competência (saber) e a nível de comportamento ele define o desempenho (saber fazer). O progresso na aprendizagem não se faz exclusivamente pela acumulação de conhecimentos, mas também pela eliminação de hipóteses falsas, de restrições inoportunas e pela substituição de procedimentos. As mudanças qualitativas incluem ainda a diferenciação e a integração de noções. As mudanças quantitativas estão ligadas aos parâmetros de acessibilidade da informação.

O aprendizado pode se dar pela ação ou por um tutorial. A descoberta e a exploração caracterizam o aprendizado pela ação. Nestas situações os fatores importantes são o feedback, a identificação dos pontos críticos da situação e dos índices que permitem evocar situações anteriores. O aprendizado por tutorial se refere às diversas formas de transmissão do saber de

um instrutor. Neste caso é importante o papel que assumem os conhecimentos anteriores como um quadro assimilador do novo conhecimento.

1.1.4 - As formas de raciocínio

O raciocínio ou inferência, é definido como a atividade mental de produção de novas informações a partir das existentes. Eles possuem duas finalidades não exclusivas; a de buscar uma coerência entre as diferentes informações e a de decidir sobre escolhas de ações. A produção de conhecimentos pode ser feita a partir de regras gerais, cuja validade é definida pela lógica formal ou a partir de regras heurísticas que podem produzir resultados nem sempre eficazes.

A inferência é dedutiva quando, partindo de uma ou mais premissas verdadeiras, chega-se a uma conclusão seguramente correta. A inferência é indutiva quando partindo-se de premissas verdadeiras chega-se à uma conclusão mais geral não necessariamente verdadeira (generalização).

As heurísticas se aplicam nas situações de resolução de problemas. Existem três classes de problemas: (i) os de indução da estrutura, onde a tarefa principal é de busca da estrutura das relações entre os dados fornecidos, (ii) os de transformação, caracterizados por uma situação inicial, situação final a ser produzida e por um conjunto de operadores que permitem transformar as situações, (iii) e os de organização, que colocam os sujeitos frente a um grupo de elementos organizados segundo um arranjo inicial sendo necessário encontrar outros arranjos que satisfaçam determinados critérios. Os problemas mistos são formados a partir da combinação de exemplares destas três classes. Uma pessoa chega a uma situação desejada se ela possui uma boa representação da situação atual, do objetivo, dos modos para alcançá-la e eventualmente algumas situações intermediárias entre a situação inicial e o objetivo. Estes fatores definem o chamado espaço-problema. Eles formam uma rede onde os nós são os estados possíveis e os arcos, os operadores. Uma sequência de pares (nós, arcos) formam um caminho dentro do espaço de pesquisa e se este caminho inclui os estados inicial e final, o problema está resolvido. O caminho mais curto define a solução ótima. Os métodos de pesquisa, ou heurísticas de caminhamento incluem os que percorrem a rede em profundidade, em largura, os prospectivos e os retrospectivos. Enquanto os métodos prospectivos baseiam-se na aplicação de operadores sobre o estado inicial para gerar novos estados, os retrospectivos partem do estado final à procura de sucessivos estados intermediários até alcançar o estado inicial. A heurística mais conhecida é denominada de "General Problem

Solver". Ela se baseia no bom conhecimento do domínio para avaliar a diferença entre o estado atual e o estado alvo e os operadores mais adequados.

A analogia é uma forma de raciocínio que se baseia em conhecimentos estocados na memória para compreensão de uma situação desconhecida. Trata-se de um tipo de raciocínio que visa a estabelecer uma relação de similaridade entre dois objetos ou situações diferentes.

1.1.5 - O controle da atividade

O controle da atividade é fundamental para a realização das tarefas e ações. Ele inclui três etapas: seleção, planificação e controle. A seleção de uma tarefa garante os recursos cognitivos necessários para a sua planificação, com a fixação de objetivos e elaboração de planos, e para o seu controle. O controle da realização inclui o controle da execução, a avaliação dos resultados das ações e a consequente avaliação da representação que se tem da situação.

O processo de seleção resulta de mecanismos motivacionais envolvendo o produto de dois fatores: a importância da tarefa do ponto de vista das motivações do indivíduo e a esperança de sucesso nesta tarefa. Este último parâmetro depende não somente da frequência de sucessos anteriores, mas também da crença que o indivíduo tem de que o sucesso está sob o seu controle. Em tarefas simplificadas a escolha se baseia na facilidade de realização. O modelo de regulação temporal considera que os fatores "importância" e "esperança de sucesso" podem variar durante a execução da tarefa. Segundo este modelo a força de uma intenção, parâmetro que evolui no tempo depende, além da importância motivacional e da competência da tarefa, também de sua urgência. Esta é definida como proximidade temporal da data limite permitida para a sua realização. A intenção escolhida é aquela para a qual a força momentânea é a mais forte.

A planificação das atividades se baseia em uma representação hierárquica de espaços abstratos. A estrutura geral do problema é representada mas os detalhes menores são abstraídos. Resolve-se o problema por refinamentos sucessivos, introduzindo-se os detalhes dos espaços abstratos dos níveis inferiores. A planificação não passa de uma hipótese de trabalho, pois ela necessita de avaliações e de ajustes constantes.

O controle da realização deve ser analisado segundo o quadro de uma arquitetura cognitiva com processadores paralelos (automáticos ou inconscientes) e sequenciais (simbólicos ou conscientes). A automação se verifica quando o controle da execução é

realizado por processadores capazes de funcionar em paralelo com o processador simbólico. A experiência profissional conduz ao desenvolvimento de automatismos, o que alivia a carga de trabalho do processador simbólico, permite a execução de ações em paralelo, e assegura uma redundância de controle. Esta última característica visa permitir que as falhas dos automatismos possam ser detectadas e corrigidas.

A avaliação dos resultados da ação é uma componente fundamental na modificação da representação que se tem do problema. Ela demanda uma reflexão sobre a ação que visa compreender uma situação para melhorar o processo de solução.

1.2 - Os Modelos Aplicados

Os modelos aplicados são reduções dos sistemas perceptivo e cognitivo humano e visam facilitar entendimento das principais características de seu funcionamento no trabalho.

1.2.1 - O modelo do processador humano

O "Modelo do Processador Humano" (Card, 1983) apresenta a cognição humana como um sistema de tratamento de informação. Este sistema compreende três subsistemas interdependentes; os sistemas sensorial, motor e cognitivo. Cada um possui uma memória e um processador. O desempenho da memória depende dos parâmetros capacidade, persistência e do tipo de informação memorizada. O do processador depende do ciclo de acesso à memória local.

O sistema sensorial consiste de um conjunto de subsistemas cada um especializado no tratamento de um tipo de estímulo. As memórias sensoriais estão ligadas à memória de curto termo do sistema cognitivo através de um filtro cognitivo. É este filtro que determina a natureza das informações a transferir entre as memórias dos dois sistemas. Quando a memória de curto termo do sistema cognitivo atinge seu limite, as informações provenientes da memória sensorial não são mais transmitidas e se degradam. A memória sensorial é aquela que recebe a informação dos diferentes órgãos sensitivos e a mantém em uma forma não interpretada por períodos de tempo muito curtos. A persistência da memória visual é da ordem de 200 μ s e de 1500 μ s para a auditiva. O ciclo de base do processador sensorial é da ordem de 100 μ s e é inversamente proporcional à intensidade do estímulo. Em termos práticos isto significa que são necessários 100 μ s para que um estímulo seja representado na memória sensorial e que a sensação de perceber algo se manifesta mais rapidamente quando o estímulo é intenso.

O sistema motor é responsável pelos movimentos do corpo. O modelo do processador humano define o tempo para a realização de um movimento baseado na quantidade de micromovimentos necessários e dos tempos teóricos para a realização destes micromovimentos. O número de micromovimentos depende das dimensões do alvo, da distância deste até a mão do indivíduo e de uma constante de erro. O ciclo local ao processador do sistema motor é de 70 μ s em média por micromovimento, já o ciclo total para efetuar um micromovimento depende do: (i) ciclo do processador sensorial para verificar a posição atual da mão, (ii) ciclo do processador cognitivo para ordenar a execução do micromovimento, (iii) ciclo do processador motor para efetuar o micromovimento.

O sistema cognitivo faz o processamento das informações estocadas na memória de curto termo e na memória de longo termo, que estoca os conhecimentos permanentes.

A memória de curto termo tem uma pequena capacidade de armazenamento, mas desempenha um papel fundamental em atividades mentais de conversação ou de raciocínio. Nela são estocadas as informações sobre o diálogo em curso ou os dados de um problema matemático, enquanto que a pessoa se concentra nos passos seguintes de sua atividade. A capacidade da MCT é de 7 ± 2 mnemas (unidades cognitivas simbólicas) que é ali mantida por meio de frequentes "rememorações". A persistência desta memória é inversamente proporcional ao número de mnemas armazenados simultaneamente: 73 s para um mnema e 7 s para três mnemas. Enquanto que na memória sensorial o esquecimento se dá unicamente de forma espontânea, na MCT ele se dá também pela substituição por novos itens de informação.

A memória permanente está organizada em estruturas associativas que armazenam o conhecimento adquirido no passado sobre um mesmo tipo de objeto ou evento. Os esquemas, como são conhecidas estas estruturas, atuam tanto no registro como na recuperação de uma informação. Quando a informação que chega é ambígua ou incompleta as informações já armazenadas na estrutura associada àquele evento ou objeto ajudam a interpretá-la e a completá-la. Esta característica é de grande importância no processo de aprendizado, sendo então necessária a identificação de qual estrutura de conhecimento pode ser utilizada para melhor assimilar o novo conhecimento. Outra característica importante é a organização hierárquica dos esquemas, que definem prioridades de acionamento, por exemplo, para aqueles ligados aos automatismos. Segundo este modelo a capacidade de armazenamento da memória de longo termo é praticamente infinita.

O processador cognitivo possui um ciclo de base de $70\mu\text{s}$ e seus princípios de funcionamento são os mesmos de um sistema de produção. Ele opera segundo o princípio de "reconhecimento-ação", segundo o qual o processador primeiramente determina as ações da memória de longo termo associadas aos mnemas da memória de curto termo. Num segundo tempo as ações são executadas provocando a modificação do conteúdo da memória de curto termo. A discriminação ou a dificuldade de recuperar uma informação está ligada ao número de informações respondendo ao mesmo índice de acesso. Finalmente o princípio da racionalidade define o comportamento do indivíduo como racional, que resulta da união entre o objetivo a alcançar, a estrutura da tarefa a realizar e dos conhecimentos disponíveis.

A racionalidade proposta pelo modelo do processador humano é rebatida por Hollnagel (Hollnagel, 1989) que propõe o termo "quase racional" para definir o funcionamento do sistema cognitivo humano. Em situações complexas, os especialistas, não conseguem explicar suas ações através de uma lógica consistente. Entre os motivos está o fato do raciocínio basear-se em modelos mentais, como os modelos operativos que são formas de representação da realidade. Também em muitas vezes as pessoas trabalham com preconceitos que influenciam desde a sua percepção até os modos de raciocínios empregados. Em consequência, nem sempre as soluções encontradas são as corretas. Em suma, as pessoas apresentam limitações quanto ao raciocínio lógico formal e têm dificuldades para integrar as informações no fator tempo, além de terem pouca velocidade e precisão. Por outro lado, apresentam excelente desempenho quanto ao reconhecimento de padrões, em trabalhar com generalizações, associações com experiências passadas e similaridades. Simplificando, o raciocínio humano é do tipo fraco em algoritmos e forte em heurísticas, com grande versatilidade para enfrentar situações inéditas.

1.2.2 - A teoria da ação

O ponto de partida da Teoria da Ação (Norman, 1984) repousa sobre a hipótese de que o indivíduo elabora modelos conceituais e que são os dados destes modelos que condicionam seu comportamento. O modelo conceitual é a representação mental que se tem da realidade, modificada e simplificada pelo que é funcionalmente significativo. Ela está intimamente ligada aos conhecimentos já adquiridos e à compreensão que se tem do problema, ampliando os elementos pertinentes e eliminando os secundários. Os modelos mentais variam de indivíduo para indivíduo em função de suas experiências passadas e evolui no mesmo indivíduo, em função de seu conhecimento do sistema. No caso de uma interface homem-computador, ela deve ser flexível o suficiente de modo a adequar-se a diferentes tipos de usuários, ao mesmo

tempo em que possa adaptar-se a evolução das características de um usuário específico durante seu processo de aprendizado com o sistema.

A teoria da ação descreve a estrutura do processo cognitivo para a realização da tarefa. Ele se inicia com o estabelecimento de um objetivo - O -, que é uma representação do estado a alcançar. A formação de uma intenção - I - resulta da comparação entre o estado atual e o estado a alcançar e é expressa segundo a especificação de uma sequência de ações - A - sobre os dispositivos físicos do sistema. Realizadas as ações, dá-se a percepção do estado corrente do sistema - E_S -, expresso por um conjunto de variáveis físicas, que é interpretado nos termos das variáveis psicológicas pertinentes na realização do objetivo - E_O -. A avaliação do estado do sistema em relação ao objetivo resulta da comparação entre E_O e O .

1.2.3 - O modelo do conhecimento semântico e sintático

Este modelo descreve a organização dos conhecimentos do usuário sobre a interação segundo os níveis semântico e sintático (Shneiderman, 1987). Os conhecimento semânticos envolvem de um lado os objetos e ações da tarefa e por outro os objetos e ações sobre o computador. Eles estão estocados na memória de longo termo e apresentam uma estrutura, seja ela declarativa ou procedural, que auxilia na tomada de decisão em situações complexas. Eles são adquiridos através de um aprendizado de conteúdo, fruto da análise de teorias gerais, de relações entre conceitos análogos e da apresentação de exemplos e de contra-exemplos. Os conhecimentos a nível de sintaxe estão ligados aos termos e convenções que o usuário deve conhecer para especificar as entradas e interpretar as saídas do sistema. Eles são de difícil aprendizado pois as formas das unidades sintáticas variam conforme os dispositivos físicos. Este tipo de conhecimento é adquirido através de repetidas utilizações de um equipamento (memorização de rota) e a não ser que sejam aplicados frequentemente eles são facilmente esquecidos. Outro problema está ligado à dificuldade de definir uma estrutura hierárquica que proporcione uma ajuda em situações complexas.

Este autor organiza a comunidade de usuários segundo seus conhecimentos semânticos e sintáticos. Assim os usuários podem ser novatos, intermitentes e frequentes. Os usuários novatos além de não possuírem conhecimento sintático, possuem poucos conhecimentos sobre a estrutura da tarefa e do computador. Os usuários intermitentes tem bom conhecimento semântico tanto sobre a tarefa como do computador, mas podem ter problemas para relembrar os detalhes do conhecimento sintático. Já os usuários frequentes apresentam bons

conhecimentos em ambos os níveis. Se o tipo de usuário for intermitente ele pode requerer algum treinamento na operação do sistema. Neste caso, as definições sobre os objetos de interação podem privilegiar menus em detrimento de linhas de comandos por exemplo.

Os projetistas de sistemas interativos podem aplicar este modelo tanto para fins educacionais como para sistematizar seus esforços. A semântica dos objetos da tarefa e do computador deve ser explicitada sempre que possível.

1.2.4 - O modelo de linguagem

| Nível de Abstração | Descrição do Sistema Interativo em termos de ... |
|--------------------|--|
| Conceitual | objetos e ações do domínio da tarefa considerados como o conteúdo do sistema |
| Semântico | significado das ações do sistema vistos como frases de um diálogo. |
| Sintático | regras para a formação de frases corretas a partir das unidades semânticas (objetos de interface). |
| Lexical | repertório de elementos não significativos que formam as unidades semânticas . |

Tabela 1.1 - Níveis de abstração do modelo da interação segundo Foley e Van Dam, 1984.

O modelo de Foley e Van Dam (Foley, 1984) descreve a interação homem-computador a partir da analogia com uma linguagem interpessoal. Assim, este modelo propõe quatro níveis de descrição: conceitual, semântico, sintático e lexical. A nível conceitual o sistema interativo é descrito pelos objetos e pelas ações de um determinado domínio da tarefa. A descrição semântica envolve os significados das ações e dos objetos do sistema. Os elementos semânticos são entendidos como grupamentos, sequências ou frases de um diálogo. O nível sintático define as regras para a construção de frases corretas especificando uma ação computacional, e para a interpretação das respostas do sistema. As entidades sintáticas que participam destas regras, formam os menores componentes significativos da interação. De sua decomposição resultam elementos sem significado. O Nível lexical descreve o sistema a partir do repertório de elementos não significativos que podem ser agrupados para formar as

unidades sintáticas. Estes incluem tudo o que se possa gerar através dos dispositivos de entrada e de saída do computador; caracteres, sons, primitivas gráficas, etc.

1.2.5 - O modelo de N. Sisson

O modelo de N. Sisson, citado em Coutaz (Coutaz, 1990), fornece uma visão integrada entre as atividades mentais dos usuários e os tratamentos do sistema nos diversos níveis de abstração. Ele integra a teoria da ação de Norman, o modelo de conhecimentos em níveis de abstração de Schneiderman e o modelo linguístico de Foley e Van Dam. O usuário desenvolve representações mentais sobre a tarefa e sobre o sistema. O sistema, por sua vez modela a tarefa e usuário. O modelo mental do usuário pode ser analisado segundo os níveis semânticos, sintáticos, lexicais e motores. A cada um destes níveis correspondem dois componentes na arquitetura do sistema: um analisador da entrada e um realizador da saída. Em última análise, a interação é decomposta em uma sequência de tratamentos semânticos, sintáticos, lexicais e motores.

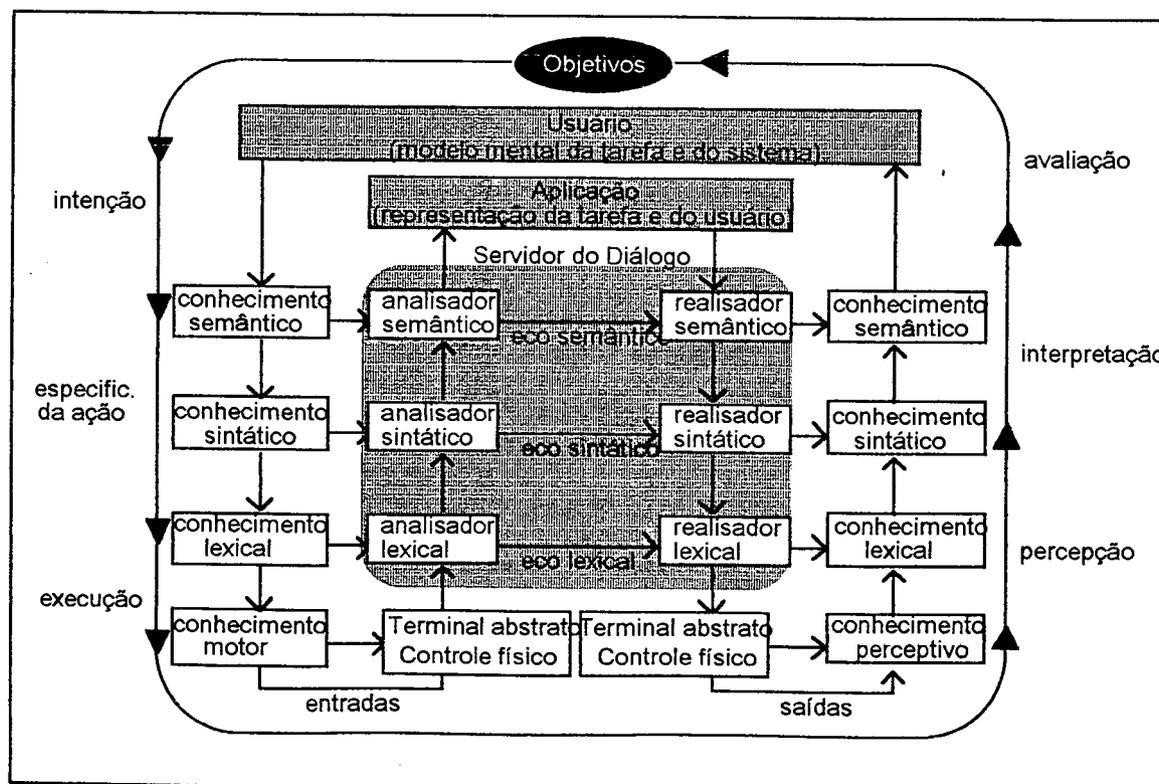


Fig 1.1 - Modelo de N. Sisson - (Fonte Coutaz, 1990)

1.2.6 - Os modelos - GOMS e Keystroke -

Os modelos GOMS e Keystroke se propõem a formalizar a realização de tarefa segundo as características do processador humano. Ambos modelos são quantitativos e procuram prever o tempo para a realização de uma tarefa através de um dispositivo informatizado. Keystroke considera somente a execução física da tarefa através dos dispositivos de entrada e saída. GOMS inclui a etapa de aquisição, durante a qual o usuário monta a representação mental da tarefa. Tratam-se de modelos redutores pois, devido ao princípio da racionalidade do processador humano, eles não admitem os erros de operação da parte do usuário.

O modelo GOMS descreve a tarefa através de Objetivos (Goals), Operadores, Métodos e Seleções. Um objetivo representa um estado do sistema a alcançar. Ele está organizado de maneira hierárquica, com objetivos complexos formados por objetivos elementares. A cada objetivo está associado um conjunto de métodos aplicáveis para a sua realização. Os objetivos elementares estão associados a uma sequência de operadores. Um operador é uma ação elementar que provoca uma mudança de estado do sistema (estado mental do usuário e/ou estado do ambiente). Um método descreve um procedimento para alcançar um objetivo e consiste de uma cadeia condicional de objetivos e operadores. As seleções se referem às escolhas entre os métodos em conflito e são expressas em termos de regras de produção. Conhecendo os tempos para a execução de cada operador é possível prever o tempo necessário para a realização da tarefa.

Este modelo descreve a interação segundo quatro níveis de detalhamento; tarefas, funções do sistema, argumentos das funções e ações físicas. A nível de tarefa o sistema é descrito como uma árvore hierárquica de sub-tarefas, cujas "folhas" são as tarefas elementares.

No modelo Keystroke não existem escolhas, o método (sintaxe) está definido de antemão (não existem seleções) e a modelagem da tarefa se dá ao nível dos operadores elementares; **Keystroking** - pressionar uma tecla do teclado ou do mouse, **Pointing** - deslocamento do cursor do mouse em direção ao alvo, **Homing** - retirada da mão para a mudança de dispositivo físico de interação (avaliado em 0,4 s), **Drawing** - ação de desenhar na tela utilizando o mouse (dependente das primitivas gráficas disponíveis), **Mental Activity** - atividade mental que o usuário realiza para preparar a execução de um operador e **Response**

Time - tempo de tratamento de um comando pelo sistema. O tempo de execução de uma tarefa é a soma dos tempos de execução de cada operador.

1.4 - Conclusões

As contribuições teóricas apresentadas neste capítulo explicam as estruturas e os processos que caracterizam o sistema cognitivo humano. Elas fornecem informações qualitativas e quantitativas que podem influenciar fortemente as decisões de projeto no tocante à quantidade, ao formato e à natureza das informações a apresentar. Os modelos da interação homem-computador, são desenvolvidos em sua maior parte numa analogia com a articulação de uma linguagem interpessoal. Assim eles propõem uma visão destas interfaces segundo os níveis de abstração conceitual, semântico, sintático e lexical.

Estes dois tipos de contribuições representam a carga de informação básica de que dispõem os projetistas para a tomada de decisão ergonômica. Apesar de pertinentes estas contribuições enfrentam diversos tipos de problemas para sua aplicação devido principalmente à extensão, organização e disponibilidade das fontes de referência.

O seguimento desta monografia apresenta as formas de intervenções ergonômicas possíveis e identifica nos ambientes computacionais de geração de interfaces homem-computador uma forma de intervenção econômica e compatível com a cultura de projetistas de informática.

2 - ERGONOMIA - AS FORMAS DE INTERVENÇÃO

Este capítulo refere-se às formas de intervenção ergonômica na concepção de interfaces homem-computador. O objetivo é de identificar uma que possa garantir um tipo de intervenção sistemática. A partir da apresentação da abordagem ergonômica para a concepção destas interfaces são analisadas as possibilidades em termos de um sistema de conformidade da qualidade ergonômica e de um ambiente computacional de geração de interfaces homem-computador. A proposta da arquitetura "Archie" para a geração de interfaces ergonômicas corresponde a uma forma de intervenção econômica e adequada à cultura de projetistas de informática.

O capítulo se encerra com a proposta inédita de um modelo de objeto de interação que facilita o raciocínio ergonômico. Os OIA^e - objetos de interação abstratos ergonômicos - são componentes lógicos da arquitetura Archie. Eles são identificados nas recomendações ergonômicas, com as quais apresentam uma correspondência efetiva.

2.1 - A Abordagem Ergonômica

A principal ressalva dos ergonomistas quanto aos métodos atuais de concepção é a de que eles se baseiam na tarefa prescrita e não na atividade real. Estes métodos privilegiam o desempenho dos sistemas que são muitas vezes dotados de todas as funções possíveis e imagináveis e desconsideram a interface e o usuário. Aspectos críticos para o sucesso da interação como o tratamento de erros e incidentes estão via de regra, ausentes.

Uma abordagem ergonômica para o desenvolvimento de sistemas interativos deve prever a introdução da análise ergonômica do trabalho e dos ciclos de prototipagens rápidas.

A análise ergonômica do trabalho apresenta duas etapas principais: a análise da tarefa e a análise da atividade. A primeira envolve o trabalho prescrito, que descreve formalmente como o usuário deve realizar sua tarefa. Deste modo descreve objetivos, procedimentos, regras de funcionamento, restrições, etc. O levantamento de dados é feito através de entrevistas, análise da circulação e do tratamento das informações, análise da organização do trabalho, das ligações entre os serviços, das características dos postos de trabalho, etc. Esta etapa descreve assim a lógica de funcionamento de um sistema. A análise da atividade, ou do trabalho realizado descreve como a pessoa executa efetivamente sua tarefa. Ela baseia-se em

observações "in loco" de seções de trabalho, onde é feito o registro das informações realmente utilizadas, de sua ordem, daquelas que faltam, das que são inúteis e das que induzem a erros. Também são colhidas informações sobre as operações efetuadas, seu encadeamento e suas dificuldades, além dos tipos, frequências, causas e condições de aparecimento dos erros e incidentes. Identifica-se assim, a lógica de utilização do sistema (Richard, 1983).

A técnica de prototipagens prevê fases rápidas de concepção e de avaliação de versões preliminares de uma interface. As avaliações destes protótipos podem ser de dois tipos: as empíricas e as analíticas.

As avaliações empíricas implicam na participação de usuários em testes de adaptação e de facilidade de uso "in loco" ou em laboratório, onde são colhidas informações sobre o seu desempenho, principalmente em termos de tempo de execução e de aprendizagem. Implicam também em julgamentos de satisfação e em análises das verbalizações por parte do usuário (Ravden & Johnson, 1989). É sem dúvida, um tipo de atividade que requer a participação de ergonomistas ou psicólogos para o planejamento, execução, tratamento e interpretação dos dados colhidos. Igualmente evidentes são os prazos incompatíveis com a dinâmica de desenvolvimento (Pollier, 1991).

A chamada abordagem de avaliações analíticas também implica na participação de especialistas em ergonomia que entretanto, buscam identificar "a priori" as características favoráveis ou desfavoráveis de uma interface (Barthet, 1988). Procura-se então, prever o desempenho dos usuários, com base no conhecimento de experts em ergonomia de sistemas interativos. Este tipo de abordagem representa uma importante economia de tempo de projeto se comparada com a abordagem empírica.

A abordagem analítica se baseia em um tipo de conhecimento prático que resulta da aplicação de teorias e modelos das ciências cognitivas às questões de projeto e avaliação destas interfaces. Eles correspondem aos critérios e às recomendações ergonômicas.

2.1.1 - Os Critérios Ergonômicos

Os critérios ergonômicos são direções a seguir nas escolhas de projeto que podem garantir um comportamento mais eficiente de parte da interface com o usuário (Scapin, 1990). Bastien (Bastien, 1991) em seu estudo para avaliar a pertinência das definições e subdivisões do conjunto de critérios ergonômicos apresenta uma lista organizada segundo três níveis de: critérios principais, sub-critérios e critérios elementares.

2.1.1.1 - Condução

A condução se refere ao conjunto de meios empregados para aconselhar, informar e conduzir o usuário na interação com o computador. Para tanto deve-se ter presteza na apresentação de uma informação adequadamente agrupada ou discriminada e fornecer feedback imediato das ações do usuário.

"Presteza: corresponde às informações fornecidas aos usuários, relativas ao estado no qual ele se encontra na interação; às ações possíveis ou esperadas e como acioná-las; às ajudas disponíveis e aos convites para as entradas de dados.

Grupamento/Distinção entre os itens:

Grupamento/Distinção por localização: se refere particularmente ao posicionamento dos itens, uns em relação aos outros, com o objetivo de indicar seu parentesco ou não parentesco a uma mesma classe, ou ainda com o objetivo de mostrar a distinção entre diferentes classes.

Grupamento/Distinção por Formato: se refere às qualidades gráficas (formato, cor, etc...) com o objetivo de indicar seu parentesco ou não parentesco a uma mesma classe, ou ainda com o objetivo de mostrar a distinção entre diferentes classes.

Feed-back Imediato: se refere às respostas do computador no seguimento das ações dos usuários. O Computador deve responder à todas as ações dos usuários o mais rapidamente possível.

Clareza A clareza se refere às características lexicais da apresentação da informação nas telas (luminância dos caracteres, contraste caractere fundo, dimensões das letras, espaçamento entre as palavras, linhas e parágrafos, comprimento das linhas, etc...

2.1.1.2 - Carga de trabalho

Se refere ao conjunto de elementos que desempenham papel na redução para o usuário de sua carga perceptiva ou mnemônica e no aumento da eficiência do diálogo. Segundo este critério as decisões de projeto devem priorizar a concisão e minimizar as ações do usuário

"Brevidade

Concisão: se refere à carga de trabalho ao nível perceptivo e mnemônico com relação aos elementos individuais das entradas e saídas.

Ações Mínimas : se refere à carga de trabalho ao nível das opções e meios disponíveis para atingir um objetivo.

Carga Mental : se refere à carga de trabalho do ponto de vista perceptivo e mnemônico para o conjunto dos elementos."

2.1.1.3 - Controle

Este critério se refere ao mesmo tempo, ao controle que o usuário tem sobre o aplicativo e sobre a interface e também, ao carácter explícito de suas ações.

"Ações Explícitas: a interface deve executar somente as operações solicitadas pelos usuários.

Controle do Usuário: se antecipar ao usuário e fornecer a ele as opções apropriadas a cada ação, permitindo que ele tenha sempre o controle da interação."

2.1.1.4 - Adaptabilidade

A adaptabilidade se refere à capacidade da interface de reagir segundo o contexto e segundo as necessidades e preferências dos diversos usuários potenciais, tanto durante seu aprendizado como na execução normal da tarefa.

"Flexibilidade: se refere aos meios disponíveis para que o usuário possa personalizar a interface e adaptá-la às suas estratégias ou habilidades no trabalho, fazendo frente assim às exigências da tarefa.

Levar em consideração a experiência do usuário: se refere aos meios que permitem que o sistema respeite o nível de competência do usuário."

2.1.1.5 - Gestão de erros

As possibilidades de evitar ou diminuir a ocorrência de erros e de corrigi-los, quando estes acontecerem depende da;

"Proteção contra os Erros: se refere aos meios disponíveis para detectá-los.

Qualidade das Mensagens: se refere à pertinência e à exatidão da informação fornecida ao usuário sobre a natureza do erro cometido e das ações a executar para corrigi-lo.

Correção de Erros: se refere aos meios disponíveis para que o usuário possa corrigir seus erros."

2.1.1.6 - Homogeneidade/Consistência

Este critério se refere ao fato das escolhas de projeto, quanto aos objetos da interfaces, serem idênticas para contextos idênticos. A homogeneidade se aplica igualmente à localização e formato como à sintaxe e à denominação.

2.1.1.7 - Significância dos Códigos

A significância dos códigos e refere à adequação expressão/objeto dos códigos empregados na interface com o usuário.

2.1.1.8 - Compatibilidade

Este critério se refere à compatibilidade que deve existir entre as características dos usuários e a organização das entradas, saídas e do diálogo.

2.1.2 - As Recomendações Ergonômicas

As recomendações ergonômicas representam a fonte de conhecimentos mais utilizada pelos ergonômistas em suas intervenções (Sousa & Bevan, 1990). Elas podem assumir diferentes formatos.

As normas são documentos oficiais concebidos por organismos de padronização, como é o caso da DIN 66234 - Principles of Dialogue Design e da ISO/CD 9241-14 - Menu Design Guidelines (ISO, 1993). Elas compõem a uma preocupação ergonômica, mas são muito gerais para uma condução efetiva e não fornecem qualquer tipo de ajuda quanto à sua aplicação. O seu descumprimento está sujeito a uma sistemática de exceção formal.

Os guias de recomendações para o projeto de IHC têm um carácter público e oferecem uma condução flexível quanto os focos de atenção e objetivos a alcançar (Bodart & Vanderdonck, 1993b; Brown, 1988; Rivlin, Lewis, & Davies-Cooper, 1990; Scapin, 1986; Smith & Mosier, 1986; Valentin & Lucongsang, 1987). Eles trazem exemplos, contra-

exemplos e comentários sobre a aplicação das recomendações, o que não dispensa porém uma interpretação detalhada para adaptá-las a uma situação específica.

As regras de projeto para um sistema ou ambiente particular constituem os chamados "guias de estilo" de grandes companhias de software (IBM, 1989; OSF, 1990). Eles tem uma circulação privada sendo o seu cumprimento uma decisão gerencial. Eles fornecem recomendações claras e favorecem a consistência das interfaces.

Finalmente as recomendações podem ser transformadas em algoritmos computacionais específicos a uma ferramenta de projeto. Eles aceleram as atividades economizando recursos e impõem uma consistência.

Os critérios ergonômicos são muitas vezes utilizados para organizar as recomendações encontradas em guias de projeto, como mostrado a seguir (Cybis, 1990):

Carga de trabalho

Não solicitar o mesmo dado várias vezes, tampouco as informações facilmente dedutíveis.

Utilizar valores "défaut."

O cursor não deve ter acesso a áreas não utilizadas da tela.

Fornecer movimentação automática do cursor.

Fornecer movimentação do cursor por tabulação.

Fornecer justificativa automática para valores decimais.

Minimizar as mudanças de modo mouse/teclado, shift/normal.

Evitar a necessidade de duplo toque.

O software deve guardar a memória da interação de modo que o usuário não tenha de entrar os mesmos parâmetros toda a vez em que for repetir um comando.

Gestão de erros.

Fornecer proteção adequada contra o acionamento involuntário de funções perigosas.

Verificar a entrada de dados para detectar erros de formato e de conteúdo

Fornecer funções UNDO para anular uma operação e recuperar os sistemas, permitindo novas entradas.

Compatibilidade.

Seguir o ritmo do usuário e não o do computador.

Quando da transcrição de dados de um documento, a tela deve ter o mesmo formato deste.

Em outros casos, seguir a sequência lógica da tarefa.

As recomendações apresentam possibilidades importantes sob o ponto de vista de um ambiente de geração de interfaces homem-computador. Porém é necessário resolver alguns problemas relacionados principalmente com a sua organização e terminologia, generalidade e/ou especificidade.

Em sua maioria, elas são conclusivas naquilo que se refere às características de sintaxe e do léxico (ver modelo de linguagem), e que caracterizam a facilidade de uso de uma interface. Por exemplo; "não solicitar o mesmo dado várias vezes" ou "o software deve guardar a memória da interação de modo que o usuário não tenha de entrar os mesmos parâmetros toda a vez em que for repetir um comando", são recomendações completas do ponto de vista do programador. Ele não precisa de informações adicionais para aplicá-las. Do ponto de vista do usuário, elas contribuem para facilitar a operação da interface.

Já as recomendações ligadas aos níveis semântico e conceitual, e que caracterizam o atributo adaptação à tarefa de uma interface, não são inteiramente conclusivas. Por exemplo: "Quando da transcrição de dados de um documento, a tela deve ter o mesmo formato deste. Em outros casos, seguir a sequência lógica da tarefa". Para aplicar este tipo de recomendação é necessário que o projetista conheça a sequência lógica da tarefa. Assim, ele deve ter outro tipo de suporte, que possa atuar a nível semântico e conceitual da interface. Do ponto de vista do usuário, esta recomendação contribui para adequar a ferramenta à tarefa.

Analisando os exemplos de recomendações ergonômicas citados neste tópico verifica-se que seu formato é especialmente apropriado ao raciocínio baseado em regras de produção. Este tipo de regra pode corresponder às primitivas da base de conhecimento de um ambiente de geração automática de interfaces homem-computador ergonômicas.

2.2 - Um sistema de conformidade ergonômica

Chapanis e Budurka (Chapanis, 1990) afirmam que a ergonomia não faz parte da "corrente" principal de engenharia pois as normas e recomendações ergonômicas não são

testáveis. Este fato torna impossível fazer a qualificação ergonômica do produto. Por outro lado as avaliações de facilidade de uso ocorrem muito tarde no ciclo de desenvolvimento, quando as principais decisões já estão congeladas.

Smith (Smith, 1986), e Chapanis e Budurka (Chapanis, 1990) argumentam que os programadores e analistas não possuem a formação para interpretar os padrões e recomendações que constituem os auxílios ergonômicos mais utilizados. Assim o fato de transferir o raciocínio ergonômico a estes profissionais acarreta um aumento de suas responsabilidades e acaba por desviá-los de suas atividades principais, de acerto dos problemas de software e de hardware.

A divisão de responsabilidades sugerida por estes autores se baseia em competências específicas. A tomada de decisão ergonômica, envolve a interpretação de normas e recomendações ergonômicas com adaptações aos casos específicos e acertos frente aos compromissos entre recomendações conflitantes. Ela é de responsabilidade dos ergonomistas que participam da concepção do sistema. Cabe a eles, face sua formação e experiência, produzir especificações ou regras de concepção de interfaces que devem ser específicas a cada novo sistema. Cabe aos programadores, profissionais habituados ao trabalho segundo especificações claras, a tarefa de seguir estas especificações.

Em vista disto eles propõem uma sistemática de especificações onde ergonomistas têm a tarefa de produzir um documento com requisitos específicos de ergonomia. Este documento deve resultar da realização da análise ergonômica do trabalho, da aplicação e da tradução de normas e recomendações à situação do projeto. Estabelecidos em uma forma verificável estes requisitos se destinam aos programadores, que se responsabilizam pelo seu cumprimento e pelos aspectos da programação de sistemas.

Por outro lado, o resultado das análises realizadas pelos ergonomistas experts em interfaces homem-computador apresentam problemas no tocante a abrangência e a variabilidade. Bastien (Bastien, 1993) mostra serem necessários um número entre 4 e 5 ergonomistas com formações distintas para que sejam identificados 80 % dos problemas ergonômicos de uma interface. Além de serem em pequeno número estes profissionais encontram barreiras internas para executarem sua missão (Faveaux & Yedid, 1991). Muitos programadores e analistas consideram que as intervenções ergonômicas como as propostas por Chapanis & Budurka freiam em demasia o ritmo do projeto. Elas exigem modificações nos

planos e metodologias já estabelecidas é necessário um trabalho em equipes multidisciplinares com reuniões de coordenação e de validação das tarefas dos diferentes parceiros.

2.3 - A proposta da arquitetura -Archie-

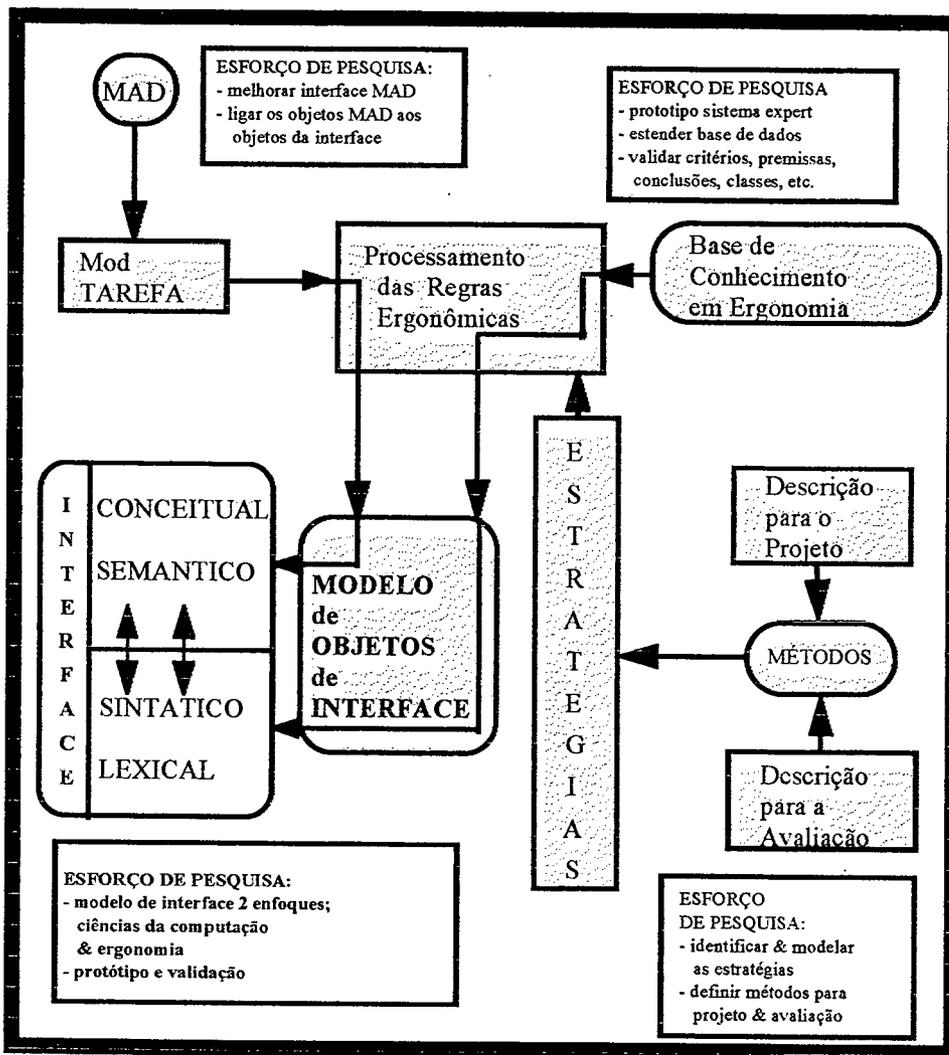


Fig 2.1 - A arquitetura Archie para a geração de IHC

Uma outra solução possível se refere à implementação de um ambiente lógico de apoio à concepção de interfaces homem-computador que traga embutidas as recomendações ergonômicas na forma de algoritmos computacionais. Segundo Smith (Smith, 1986) os projetistas de informática são favoráveis à este tipo de condução que representa um meio poderoso de economizar tempo e dinheiro. A automação do projeto lhes parece a evolução óbvia para a tecnologia na área da engenharia de software.

Quando as etapas da abordagem ergonômica são examinadas sob o ponto de vista de ferramentas de apoio, verifica-se que nem a análise ergonômica do trabalho nem as atividades

de avaliação dispõem de qualquer tipo de apoio lógico. Por outro lado, nenhum ambiente de concepção propõe facilidades para tratar dos aspectos ergonômicos das interfaces.

Ao fazer a proposta para a arquitetura lógica de um ambiente lógico de apoio à concepção ergonômica, Scapin (Scapin, 1990e) considera que os usuários destas ferramentas não tenham um treinamento específico em ergonomia. Assim ela se baseia no raciocínio ergonômico implementado através de algoritmos computacionais derivados dos princípios práticos ergonômicos (critérios e recomendações).

A ação de pesquisa Archie (Advice Giving Routines for Computer-Human Interaction Ergonomics), na qual está inserido este trabalho de tese, possui quatro componentes principais: modelo da tarefa, modelo de objetos de interface, base de regras ergonômicas e base de estratégias de projeto e de avaliação. O modelo da tarefa descrito segundo o formalismo MAD, representa a entrada do sistema e as premissas para o raciocínio ergonômico. As conclusões deste raciocínio se reportam a um modelo de objetos de interface. O disparo das regras armazenadas na base de dados ergonômica é gerenciado pelos métodos e estratégias de projeto e de avaliação.

Este ambiente pretende fornecer ferramentas de apoio às atividades de ergonomistas e programadores, de modo a cobrir e integrar todas as etapas envolvidas na concepção de interfaces homem-computador segundo a abordagem ergonômica. Nos próximos parágrafos serão descritas as atividades relacionadas com as quatro linhas de pesquisa definidas para o projeto

2.3.1 - O modelo da tarefa.

A abordagem ergonômica para a concepção de um sistema interativo impõe a análise ergonômica do trabalho. Sob o ponto de vista de um sistema de concepção é necessário a montagem de um modelo da tarefa para fornecer as informações envolvidas com o disparo das regras ergonômicas. O ponto de partida para se chegar a este modelo reside nas técnicas de análise direcionadas a uma formalização da tarefa segundo os objetivos do operador (Sebillotte, 1991). O modelo que resulta desta análise deve ser capaz de evidenciar a organização da tarefa em termos de objetivos e sub-objetivos; as relações lógicas (e/ou) e de sincronização (antes/depois) entre as sub-tarefas, a demanda de informação relativa a cada sub-tarefa e o vocabulário empregado. O formalismo denominado *Méthode Analytique de Description - M.A.D* - está sendo desenvolvido de maneira a permitir a descrição destes

aspectos, fundamentais face ao raciocínio ergonômico (Scapin, 1989) O primeiro protótipo de uma ferramenta de edição visando a descrição da tarefa já foi implementado (Pierret-Golbreicht, Delouis & Scapin, 1989). A partir dos dados entrados segundo o formalismo MAD a ferramenta constrói a representação gráfica da decomposição da tarefa segundo a hierarquia e as relações lógico-temporais descritas. Isto permite validar a descrição de uma forma interativa entre usuário e analista. Resta ainda definir a montagem da estrutura de dados sobre os objetos da tarefa, que deve ser investigada em suas inter-relações com os objetos de interface e seus atributos.

2.3.2 - A base de conhecimento em ergonomia

A análise das contribuições das ciências cognitivas às questões de projeto de interfaces com o usuário afasta as possibilidades de teorias e modelos contribuírem na formação de uma base de conhecimento ergonômico para um ambiente de apoio. Por outro lado Scapin identifica problemas devido a abrangência limitada, ao caráter geral e à falta de robustez de recomendações ergonômicas originárias de experimentos de laboratório. Ele considera que um sistema de apoio possa também ter sua abrangência limitada e que ferramentas de apoio no esclarecimento das recomendações e na identificação do problema possam ser desenvolvidas. Assim, a construção da base de conhecimento do projeto Archie se fundamenta nas recomendações ergonômicas.

O primeiro passo já foi dado e consistiu na coleta de aproximadamente mil recomendações a partir da bibliografia especializada. Esforços futuros devem visar sua classificação segundo os critérios ergonômicos e segundo os objetos de interface, além da homogeneização do vocabulário. Resta também indefinida a questão do formato de regra de produção a ser utilizado na tradução das recomendações. Um estudo preliminar sugere o seguinte formato:

SE (premissa) USANDO (Critério) ENTÃO (Conclusão)

As premissas das regras incluem o tipo de usuário (novato, expert, ocasional, etc.), o tipo da tarefa (entrada de dados, edição de texto, busca de informação, etc.), tipo de atividade de projeto (projeto de tela, de títulos, de rótulos, etc.) e o contexto (quantidade de informação já existente na tela). Os critérios são aqueles definidos por Scapin e Bastien (Bastien, 1991) e incluem a condução, a carga de trabalho, o controle, a adaptabilidade, a gestão de erros, a significância dos códigos, a consistência e a compatibilidade. As conclusões podem ser

relativas a uma nova atividade de projeto, à demanda de novas informações ou a uma conclusão sobre a avaliação (relatório com as regras violadas).

2.3.3 - Os métodos e estratégias de projeto e de avaliação

As pesquisas nesta área visam identificar as estratégias e os critérios utilizados no disparo de regras ergonômicas. Para tanto Pollier investigou em situação real a atividade de especialistas na avaliação dos problemas ergonômicos de uma interface (Pollier, 1991). Esta autora identifica um elenco de estratégias e sugere o seguinte encadeamento; com alguma variação os especialistas iniciam o processo de avaliação seguindo os objetivos da aplicação, ou seja, ele identifica as principais tarefas e sub-tarefas que o usuário deve realizar em seu trabalho. Estas sequências de telas são analisadas então segundo os níveis de abstração (conceitual, semântico, sintático e lexical). Em cada nível o analista identifica as classes de objetos de interface existentes sobre os quais ele aplica os critérios e as recomendações ergonômicas.

2.3.4 - O modelo de objetos de interface.

É sobre o modelo de objetos de interface que se dá o disparo das regras derivadas das recomendações ergonômicas. Estes objetos integram tanto as premissas como as conclusões do formato de regra definido no parágrafo anterior.

Como será descrito no próximo capítulo, a interface de um sistema interativo é determinada por dois tipos de entidades computacionais : os objetos de interação e os objetos do controle do diálogo. Os primeiros, também conhecidos como objetos de apresentação, representam a dimensão estática da interface. São eles que interagem diretamente com o usuário, recebendo suas ações e apresentando dados e informações. Os objetos do controle do diálogo são responsáveis pelo aspecto dinâmico da interface. Eles controlam a sequência de ações determinando quais são os objetos disponíveis para a interação em um determinado momento. Os objetos do controle do diálogo tem um caracter específico, estando fortemente ligados às características de cada aplicativo.

De maneira a garantir a viabilidade deste trabalho, o foco das atenções é centrado sobre os objetos de interação. Estes objetos integram as chamadas caixas de ferramentas que estão disponíveis nas diversas plataformas para o desenvolvimento de interfaces homem-computador. Para se chegar até eles são identificadas duas abordagens possíveis. A primeira examina exatamente os ambientes de concepção e a segunda, as recomendações ergonômicas.

A questão que se coloca frente à primeira abordagem diz respeito à portabilidade das configurações realizadas por um sistema de geração de interfaces face às diferentes plataformas de programação existentes. Uma vez que eles são construídos a partir dos recursos de um sistema de janelas específico (X-Windows, MS Windows, Finder), cada plataforma propõe objetos de interação distintos. As soluções possíveis incluem a definição de um conjunto de objetos resultante ou da união (Chauvet, Harman, & Jouhier, 1991) ou da intersecção (Bodart & Vanderdonckt, 1993a) dos objetos existentes em cada ambiente. Enquanto que a primeira solução garante a portabilidade de uma interface pela exaustão a segunda o faz pela abstração. A primeira lida com objetos concretos e a segunda com objetos abstratos. Um objeto de interação concreto possui ao menos um correspondente nos diferentes ambientes de concepção. Um objeto de interação abstrato pode não ter correspondente algum em um determinado ambiente. Neste caso ele deve ser construído pelo projetista.

Outra dificuldade encontrada por esta abordagem diz respeito ao raciocínio ergonômico. A correspondência entre os objetos de interação concretos ou abstratos, seus atributos e as recomendações ergonômicas não é absolutamente efetiva. Verifica-se que estas recomendações referem-se a componentes de interfaces distintos dos objetos de interação encontrados nas caixas de ferramentas. De fato, os objetos trazidos por estas recomendações existem nos modelos mentais dos ergonomistas e suas características principais referem-se à lógica de utilização do sistema. A adoção de um modelo de objetos de interação concretos ou abstratos não é favorável ao raciocínio ergonômico pois seus atributos estão voltados para a lógica de funcionamento do sistema.

A segunda abordagem possível para a definição do modelo de objetos de interação é baseada no exame de recomendações ergonômicas. Através dela é possível identificar as classes de componentes de interface que os ergonomistas consideram em suas recomendações. Estes componentes favorecem o raciocínio ergonômico na medida em que a correspondência entre seus atributos e as regras ergonômicas passa a ser efetiva. Eles representam a proposta deste trabalho de tese e correspondem à noção de OIA^e - objetos de interação abstratos ergonômicos -. Um OIA^e define uma abstração cujo objetivo vai além da independência em relação aos diversos ambientes de concepção. Ele se propõe também a facilitar o raciocínio ergonômico sobre os aspectos ligados à utilização de um sistema: o significado dos objetos e as relações entre eles.

A proposta dos OIA^e necessita no entanto de um esclarecimento maior sobre a natureza destes componentes, que não é a mesma dos objetos de interação usuais. Sua morfologia em termos de atributos, voltada para a lógica de utilização carece de uma definição precisa.

A perspectiva teórica aplicada para a construção de um OIA^e vem da Semiótica, a ciência que estuda os sistemas de sinais e sua vida em sociedade. Como será visto no capítulo III, um sinal corresponde à associação entre um plano de conteúdo e um plano de expressão que ocorre na mente de um interpretador (Andersen, 1990). A definição de um OIA^e busca uma analogia com um sinal e se baseia na morfologia computacional de um agente PAC (Coutaz, 1990) para definir uma entidade lógica tridimensional; atributos de conteúdo e de expressão mantidos coerentes através de um plano de controle. Esta entidade resulta no que Andersen chama de sinal candidato, pois seu conteúdo acontece na mente do programador e não na do interpretador do sinal, no caso o usuário. Seu mérito entretanto reside exatamente no fato de explicitar os conteúdos pretendidos pelos projetistas, que passam a fazer parte dos atributos configuráveis de uma interface homem-computador.

2.4 - Conclusões

Este capítulo traz a proposta de um ambiente computacional para a geração automática de interfaces homem-computador, cuja base de conhecimento é derivada das recomendações ergonômicas. Este tipo de intervenção não dispensa a realização da análise ergonômica do trabalho nem dos ciclos de prototipagens rápidas. Sua finalidade, de apoiar a concepção de versões de interfaces ergonomicamente corretas, pode garantir entretanto, uma maior objetividade no processo de desenvolvimento. Além de justificar-se sob o ponto de vista econômico, um ambiente computacional de geração de interfaces conta com a simpatia dos principais agentes do processo de concepção. Os projetistas de informática consideram-no como evolução óbvia para a tecnologia na área de engenharia de software.

É feita a proposta inédita de objetos de interação abstratos ergonômicos -OIA^e, que têm o objetivo de facilitar o raciocínio baseado em regras de produção derivadas das recomendações ergonômicas. Eles compõem o modelo de objetos que integra o ambiente de geração de interfaces pretendido.

A definição de OIA^e, suas classes e seus atributos são apresentados na segunda parte desta monografia. Seu entendimento necessita porém do esclarecimento dos conteúdos

apresentados nos capítulos III e IV. Eles referem-se às interfaces homem-computador vistas a partir de duas perspectivas: a computacional e a semiótica.

3 - IHC - PERSPECTIVA COMPUTACIONAL

Este capítulo trata das interfaces homem-computador a partir do ponto de vista de seu funcionamento. Assim, são apresentados os componentes funcionais, as arquiteturas e as ferramentas disponíveis para sua concepção de sistemas interativos.

O termo Sistema Interativo define um sistema computacional cujo processo está acessível a um usuário, ou operador de um grande número de serviços de informação. O tipo de sistema aqui tratado reflete a tendência atual da tecnologia em termos interfaces gráficas interativas. Deste modo, ele inclui os seguintes módulos funcionais: aplicação ou núcleo funcional, controle do diálogo, objetos de interação, sistema de janelas e dispositivos físicos.

A distinção entre os aspectos de conteúdo e apresentação, caracteriza os principais modelos de arquitetura discutidos neste capítulo. Os sistemas construídos segundo esta arquitetura reagrupam de um lado as funções específicas do domínio e de outro aquelas especializadas na gestão dos detalhes da apresentação do sistema.

As ferramentas de engenharia de software podem ser divididas em três categorias: as caixas de ferramentas, os esqueletos de aplicação e os geradores de interface. As toolboxes são bibliotecas de procedimentos ou objetos que servem de base para a construção de programas reutilizáveis e extensíveis que estruturam uma aplicação. Os geradores automáticos de interface baseiam-se nos esqueletos de aplicação e em um sistema de especificação adequado.

3.1 - Os componentes funcionais

A figura 3.1 mostra os componentes funcionais presentes em todo o software gráfico e interativo (Bass & Coutaz, 1991). O aplicativo representa os conceitos e as funções que modelizam um grupo de competências do domínio. O núcleo funcional não deve ter conhecimento de como suas estruturas de dados e funções são apresentados aos usuários. Entretanto, ele deve sempre saber com qual usuário está interagindo e quais são suas características. O controle do diálogo define quais os objetos devem participar da interação e qual o seu sequenciamento. Assim ele determina tanto o estilo do diálogo como a lógica de funcionamento do sistema. Os objetos de interação definem entidades que o operador pode perceber e manipular através do mouse e do teclado. Eles possuem os recursos para fazer a apresentação de dados e a recepção das ações dos usuários. Eles assim estão associados à

lógica de utilização dos sistemas. O sistema de janelas gerencia os recursos dos terminais abstratos e controla os dispositivos físicos. Ele proporciona a independência em relação aos serviços dos terminais físicos.

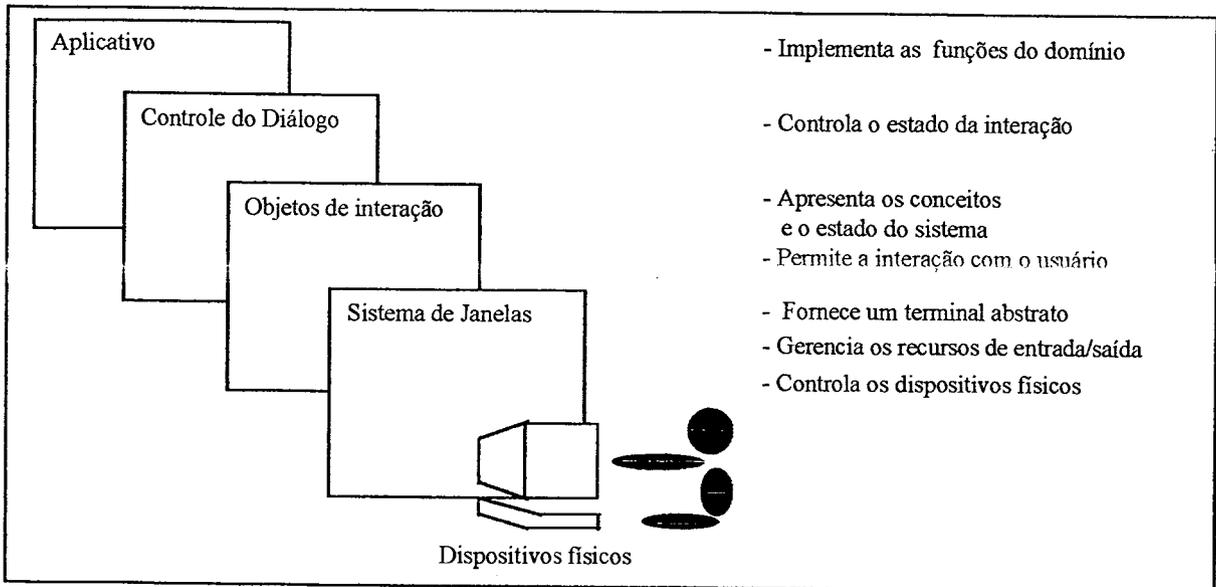


Fig 3.1 - Os componentes funcionais de um software interativo. (Fonte Ben & Coutaz, 1991)

3.1.1 - Núcleo Funcional

A definição do núcleo funcional resulta de uma clara identificação do problema, da modelagem do operador e da realização da análise do trabalho. Para fornecer o apoio que o usuário necessita na realização de sua tarefa o projetista de um sistema informatizado deve poder identificar suas características gerais e específicas. Quando o contato direto com os usuários do futuro sistema for possível o levantamento de suas características específicas pode ser feita através de entrevistas, questionários e de observações in loco. Nestas observações o usuário deve ser convidado a verbalizar sobre suas intenções e o sobre o desenrolar de suas ações. Este contato é possível nos casos em que o novo sistema substitui sistemas ou métodos já existentes. As características gerais dos usuários podem ser conhecidas através dos modelos teóricos e aplicados sobre o funcionamento do sistema cognitivo humano conforme apresentados no capítulo I. Eles representam a única fonte de informação no caso de um sistema inédito onde o contato com o usuário não é possível.

O objetivo da análise do trabalho é principalmente o de estruturar o sistema de acordo com o modelo mental do usuário. O resultado desta análise consiste em um conjunto de tarefas e de conceitos que o usuário deve executar e manipular para alcançar seus objetivos. De posse

destes dados o projetista deve realizar a decomposição da tarefa em sub-tarefas como maneira de estabelecer a granulometria e a hierarquia das funcionalidades do sistema. O usuário pode ter problemas de controle se as funcionalidades forem muito poderosas ou ao contrário, ter uma carga de trabalho elevada com funcionalidades de baixo nível. Em especial a atenção deve ser centrada nas questões envolvendo tarefas alternativas e sub-tarefas que possam ser executadas em paralelo ou em qualquer ordem. A hierarquia de funções define o aspecto dinâmico da interação e a flexibilidade geral de uma interface.

O projeto do componente funcional inclui também os conceitos envolvendo a gestão do posto de trabalho. No caso de um sistema multi-tarefa o domínio do gerente do posto de trabalho envolve o lançamento, a terminação de aplicações e a determinação de qual aplicação é a corrente. Este é o papel de uma "shell" Unix e dos "desktop managers".

Em última análise o projeto funcional determina o diálogo em um sistema interativo.

3.1.2 - Controle do diálogo

A principal função do controle do diálogo é a de controlar o comportamento dinâmico da interface. Esta função pode ser expressa de uma outra forma, qual seja, a de manter a correspondência entre o estado do domínio e o estado da interface. Ele o faz através da manipulação de protocolos de comunicação com o núcleo funcional e com a apresentação. O termo "apresentação" designa os serviços fornecidos pelos objetos de interação e pelo sistema de janelas. Estes dois protocolos são necessários uma vez que aplicação e apresentação utilizam diferentes técnicas para representar o mesmo sistema. Para fazer a correspondência entre estes dois componentes o controle do diálogo deve poder se comunicar com seus interlocutores em suas respectivas linguagens. Os objetos da aplicação apresentam um nível de abstração semelhante aos do domínio da tarefa e desconhecem a maneira como seus dados são apresentados ao usuário. A apresentação manipula entidades cuja linguagem se aproxima das dos dispositivos físicos de entrada e de saída.

As correspondências entre estes protocolos são em sua maior parte, específicas a um sistema em particular. Entretanto, existem relações entre objetos de interação que independem do núcleo funcional e que são gerenciadas pelo controle do diálogo. Elas definem uma porção reutilizável do controle do diálogo que pode ser implementada no que se chama de esqueleto de aplicação. Um sistema interativo pode ser construído a partir da extensão e do preenchimento dos vazios de um esqueleto disponível a diversos núcleos funcionais.

Os serviços genéricos de um esqueleto incluem facilidades para macro-comandos, anular-repetir (*undo-redo*), cortar-colar (*cut-and-paste*), ajuda imediata (*help on-line*) e valores *default*. Os macro-comandos minimizam o número de ações físicas necessárias para especificar expressões de entrada. A possibilidade de anular e repetir comandos permite uma recuperação fácil e direta dos erros. Ela incentiva a exploração do sistema ao liberar o usuário da definição de uma estratégia para retornar ao estado anterior ao erro. As facilidades de cortar-colar contribuem para minimizar as ações ao permitir que o usuário reutilize as saídas do sistema como novas entradas. A ajuda *on line* é uma facilidade importante na exploração e na recuperação de erros que proporciona aos usuários os conhecimentos declarativos e procedurais sobre determinada funcionalidade. A definição de valores default auxilia nas especificações ao diminuir a carga mnemônica, o número de ações que o usuário deve realizar e as possibilidades de erros. Eles podem ser estáticos ou evoluírem durante uma sessão de trabalho e sua definição deve ser um compromisso entre as escolhas mais prováveis e as menos perigosas. A escolha do valor default para ações destrutivas deve cair sempre sobre sua negativa e não sobre a própria ação.

A especificação do controle do diálogo envolve a definição de quem deve controlar a interação. Em um primeiro nível de análise as opções se limitam ao usuário e ao sistema. Na condução pelo sistema o usuário é levado de sub-tarefa em sub-tarefa sem poder refazer suas estratégias. A condução pelo usuário permite o trabalho em diversas sub-tarefas paralelamente. Em geral o usuário deve ter a iniciativa do diálogo, mas quando ele é novato o sistema pode sugerir os sucessivos passos da tarefa. Assim o sistema interativo pode ser definido tanto como ferramenta de trabalho como colaborador, suprimindo as deficiências do usuário. Sob um outro ponto de vista o controle pode ser interno, externo ou misto. No controle interno o núcleo funcional dirige as ações, definindo a condução pelo sistema. Quando o controle do diálogo tem a iniciativa das ações a interação é controlada externamente pelo usuário. O controle é misto quando ele pode ser revezado entre núcleo funcional e controle do diálogo. Neste caso o usuário comanda a interação durante a entrada dos parâmetros de um comando cabendo ao sistema o controle na execução de ações prolongadas. Ao nível dos objetos de interação a repartição do controle permite ao usuário agir sobre diversos objetos de forma oportunista. Ele pode agir sobre um objeto, abandoná-lo temporariamente para agir sobre outros objetos e retornar ao objeto inicial em seguida.

A descrição do diálogo é um dos principais problemas na especificação de uma interface. Os modelos incluem gramáticas formais, regras de produção e diagrama de estados.

As gramáticas formais consideraram o diálogo homem-computador como uma linguagem sujeita a regras de sintaxe. O formalismo B.N.F. (Backus-Naur Form) permite a descrição de um sistema através de frases sintaticamente corretas. O formalismo C.G.L. (Command Language Grammar) é um sistema de regras aplicado as linguagens de comando que é descrito segundo os níveis conceitual, semântico, sintaxe e de interação. A descrição a nível conceitual envolve as regras sobre os objetivos e sub-objetivos. A nível semântico são descritas as regras que se aplicam aos objetos da aplicação. A nível sintático é descrita a sintaxe dos comandos e o nível de interação está ligado às ações físicas para especificar os comandos.

A descrição segundo regras de produção (se-então) se baseia em condições e ações. Em um sistema especificado por regras de produção o controle do diálogo deve monitorar uma grande quantidade de dados de modo a decidir qual regra a aplicar. A descrição segundo este formalismo induz uma perspectiva em termos de paralelismos. O problema dos formalismos baseados em gramáticas ou regras de produção está ligado à grande quantidade de frases necessárias para explicitar todas as possíveis ações do usuário.

Os diagramas de transição de estados descrevem os diferentes estados do sistema e as ações que permitem a transição de um estado a outro. Os estados são representados por nós e as ações que permitem a passagem de um estado a outro são expressas através de arcos ligando os nós. As únicas ações possíveis em um estado são aquelas relacionadas no nó. O problema das descrições segundo este formalismo está na complexidade que assumem diante de comportamentos relativamente simples. A introdução de níveis de abstração e de recursividade em formalismos estendidos visam atenuar estas dificuldades.

3.1.3 - Objetos de interação

Um objeto interativo é uma entidade computacional geralmente inspirada em alguma entidade do mundo não informatizado. Ele utiliza os serviços dos sistemas de janelas para permitir a interação com o usuário e apresentar os conceitos e o estado do sistema. Os objetos de interação estão organizados em toolkits e são especializações da noção de janelas. Eles apresentam partes como bordas, plano de fundo e primeiro plano e propriedades como posição, dimensões, coloridos, texturas, textos, imagens, etc. Em resposta às ações do usuário, os objetos de interação podem realizar outras ações ou gerar eventos de repercussão interna.

Seu comportamento se manifesta nas mudanças em suas propriedades ou nas de outros objetos. Em geral o seu comportamento imediato de um objeto não pode ser alterado nem pelos clientes nem pelos usuários. Seus atributos ou propriedades podem ser especificados em um arquivo de recursos que é lido pela toolkit antes de criação do objeto. Estes arquivos variam em complexidade conforme os ambientes. Assim o ambiente de programação Macintosh permite a especificação de objetos simples e compostos através de um arquivo de recursos chamado ResEdit. Este ambiente também proporciona facilidades para a criação, a especificação e a destruição de objetos através de manipulação direta. Os arquivos de recursos dos ambientes X podem ser gerados através da compilação de instruções escritas na linguagem UIL (User Interface Language/OSF).

Os objetos de interação seguem os princípios do paradigma de programação orientada a objetos, encapsulando dados e funções. Eles são organizados em hierarquia de classes que permitem reaproveitamento de recursos através da relação de heranças. Uma classe é uma matriz a partir da qual são produzidos exemplares de propriedades semelhantes. Um objeto é uma instância de classe, seus dados são denominados atributos e suas funções de métodos. Os atributos são descritos de forma declarativa e o métodos de forma procedural. A noção de objeto subentende ainda a existência de condições (pré e pós-condições) para a execução de métodos o que permite a verificação automática do bom funcionamento do exemplar. Os objetos são definidos a partir de um comportamento externo ou aparente. A maneira como ele realiza este comportamento é totalmente escondida das outras classes de objetos. Os objetos se comunicam por intermédio de mensagens que contêm um destinatário, um método do destinatário e os parâmetros destes métodos. Um objeto não pode ser manipulado diretamente, ele apenas atende solicitações de outros objetos. Através do mecanismo de herança cada subclasse herda os métodos e os atributos de uma ou mais classes mãe. O primeiro caso define uma herança simples e o segundo uma herança múltipla. Estes métodos e atributos podem ser estendidos ou mesmo reescritos pela subclasse. A reutilização e a especialização proporcionam o aumento de confiabilidade e da produtividade da tarefa de programação. Algumas classes da hierarquia são definidas apenas para fins de formação de heranças. Os métodos e atributos das chamadas meta-classes não definem uma entidade existente no domínio, mas são comuns a diversas outras classes que efetivamente se realizam.

O ciclo de vida de um objeto de interação inclui as etapas de criação, apresentação, ativação, vida ativa, desativação, retirada e destruição (Sacre, Sacre-Provot, & Vanderdonckt,

1992). Durante sua vida ativa um objeto está sujeito a um processo de modificação de seus atributos, seja pelo programa cliente seja pelo próprio usuário.

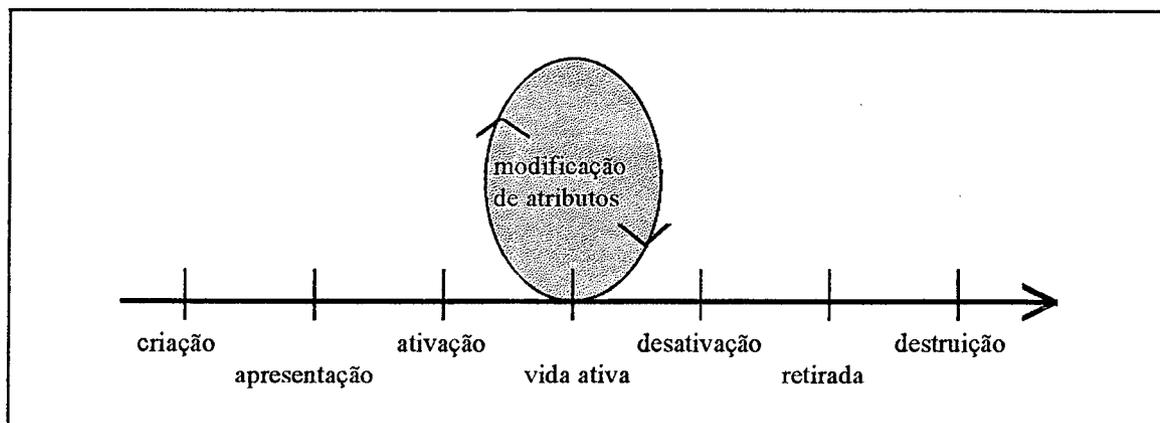


Fig. 3.2 - Ciclo de vida de um objeto de interação. Fonte (Sacre et al, 1992)

Os objetos de interação podem acionar ações do núcleo funcional e do gerente de janelas, a partir da ocorrência de determinados eventos. A cada classe de eventos corresponde uma ação da aplicação.

Estes objetos podem ser considerados como semânticos ou sintáticos. Os objetos ditos semânticos estão ligados diretamente ao núcleo funcional. Eles apresentam alguma variável do domínio da tarefa. As considerações sobre este tipo de objeto envolvem as noções de correspondência direta e de distância semântica. A falta de correspondência entre objetos computacionais e não computacionais aumenta a distância semântica. Neste caso, maior será o esforço mental que o usuário deve fazer para determinar as entradas e interpretar as saídas do sistema.

Os objetos sintáticos independem da tarefa, estando ligados especificamente à operação do sistema. A movimentação ou o redimensionamento de janelas usando a barra de título ou suas bordas vão acionar o gerente de janelas. No caso dos objetos sintáticos, o princípio básico a aplicar é o de consistência.

3.1.4 - Sistema de janelas

Um sistema de janela inclui os serviços para gerar e gerenciar terminais abstratos e controlar terminais físicos.

A independência em relação aos diversos tipos de terminais ou dispositivos físicos existentes interessa tanto do ponto de vista da portabilidade do software como da facilidade de

programação. Esta característica interessa em especial ao programador que não precisa conhecer as sintaxes específicas de um determinado terminal. Um terminal abstrato fornece um conjunto de operações virtuais predefinidas que são traduzidas nos termos das operações associadas a um determinado terminal físico. Estas operações são transmitidas ao dispositivo específico por intermédio de um controlador de terminal físico. Este último componente é que deve ser trocado quando um sistema de janelas é instalado em outro equipamento. O compromisso associado a tal solução técnica fica por conta de pequenas distorções de imagens.

Um terminal abstrato corresponde a uma janela que é associada ao processo de um aplicativo. Ela não precisa estar limitada ao tamanho da tela física. A imagem a ser apresentada pelo sistema de janelas é armazenada em superfícies de apresentação que podem ser maiores do que a tela física. Neste caso elas são chamadas de superfícies de apresentação virtuais. Uma janela é uma superfície de apresentação projetada na tela. No caso de superfícies virtuais a informação é mantida em sua escala original, sendo fornecida ao usuário a possibilidade de navegar dentro deste espaço usando as barras de rolamento. O redimensionamento da janela pode ser acompanhado ou não do redimensionamento da superfície de apresentação. Conforme o sistema, uma janela pode ser elementar, formada por uma única superfície de apresentação ou composta. Neste caso ela é formada por uma sub-janela que define o seu fundo e por outras que definem seu quadro (barra de título, ícones de fechamento, redimensionamento, barra de rolamento, etc.). Através de janela de conteúdo o usuário se comunica com o programa aplicativo. Através do quadro ele se comunica com o gerente de janelas.

Todo o sistema de janelas é gerenciador do recursos associados aos terminais abstratos: tela de alta resolução, teclado, dispositivo de indicação (mouse, joysticks, trackballs, etc.), mapeamento de cores e fontes.

Um sistema de janelas gerencia múltiplos terminais abstratos, assim ele deve fornecer ao usuário a possibilidade de diferenciar qual a janela deve receber suas ações. Os eventos do usuário são atribuídos à janela que possui o foco da atenção. Esta janela se diferencia por sua barra de título escurecida e pelo cursor de texto, apresentado como um retângulo sólido. Este foco pode ser alterado pelo usuário, valendo-se do cursor do mouse ou pelo programa cliente, quando da apresentação de caixas de diálogos.

Na apresentação de múltiplas janelas os problemas de gerenciamento envolvem o posicionamento, com a sobreposição ou a retirada de janelas. A sobreposição se baseia em uma lista de janelas ativas, cada janela acompanhada de seu tamanho e posição. As janelas na base

da lista são as que estão em primeiro plano na tela. As operações possíveis sobre estas janelas são criar, deletar, mover para o topo e mover para a base. As janelas podem ser ainda iconificadas ou abertas (de-iconificadas). As saídas podem ser enviadas às janelas escondidas sem que o usuário perceba. Assim, quando uma janela que estava escondida é mostrada ela deve ser redesenhada. Isto pode ser feito pelo cliente através de uma solicitação do sistema de janelas ou pelo próprio sistema de janelas que mantém uma cópia atualizada da janela em um *buffer* separado. Outro problema gerado pela multiplicidade de janelas se refere às relações de hierarquia entre elas. Estas relações se manifestam nas restrições de layout onde uma janela "filha" é ligada à janela "mãe", permitindo uma movimentação conjunta. O exemplo típico é fornecido pelos menus, formados como um conjunto de sub-janelas residindo na mesma janela mãe. Em alguns sistemas as barras de título e de rolamento são também implementadas como janelas filhotes. Esta hierarquia está também presente na propagação de eventos, que são enviados primeiro à janela mãe e depois às filhotes.

O chamado contexto gráfico define os aspectos de aparência de uma janela. Eles incluem as definições de cores e fontes e podem ser locais à cada janela ou globais e repartidos entre várias delas.

As diferenças na aparência e no comportamento dos diversos sistemas de janelas levam à ações de normalização que resultam na definição de estilos. Assim a Apple possui definido um guia de estilo que todas as aplicações desenvolvidas para Macintosh devem seguir. Esta empresa força o cumprimento de suas recomendações através de um sistema de certificações. Como resultado um usuário Macintosh encontra políticas de manipulação idênticas em qualquer aplicação que utilize. Os estilos para sistemas construídos sobre Unix incluem Open Look (da AT&T e Sun Microsystems) e Motif (da Open Software Foundation). Para PC existem Windows (MS-DOS) e Presentation Manager (OS-2).

3.2 - Modelos de arquiteturas para os componentes funcionais

O projetista diretamente e o usuário indiretamente, são os favorecidos por uma definição de arquitetura de software interativo. O primeiro pode se valer de uma estrutura genérica embasando a construção do sistema do qual o usuário pode tirar um proveito maior. Esta estrutura descreve o fluxo de dados entre o usuário e a aplicação, identificando as etapas de transformação de dados e definindo os responsáveis por estas transformações.

Coutaz (Coutaz, 1990) argumenta que o problema da organização do componentes funcionais de um sistema interativo está situado dentro de um espaço de quatro dimensões: aplicação, usuário, ambiente e níveis de comunicação. A estas pode ser adicionada a dimensão projetista. Cada uma delas possui suas propriedades e necessidades particulares que um modelo de arquitetura deve tentar satisfazer. A aplicação necessita de portabilidade, robustez e de manutenção. O projetista reivindica modelos e ferramentas evoluídos em suas abstração e serviços. O usuário tem o direito à condução, à proteção contra erros e à iniciativa do diálogo. O ambiente impõe limitações de ordem física. A comunicação entre os componentes funcionais necessita de concisão e de coerência.

A primeira arquitetura a ser apresentada se baseia no modelo cliente-servidor para a definição das máquinas abstratas que constituem o sistema interativo. Um segundo tipo de arquitetura busca na analogia entre uma linguagem humana a definição dos níveis de abstração que organizam um software interativo. Finalmente a arquitetura multi-agente define o sistema interativo como um conjunto de agentes reativos à estímulos.

3.2.1 - Modelo de máquinas abstratas

O modelo de máquinas abstratas é caracterizado por uma hierarquia de servidores e clientes. Uma determinada camada, denominada servidor presta serviços para camadas superiores, denominadas clientes. Assim, os clientes possíveis de um sistema de janelas são objetos de interação, o controle do diálogo ou diretamente uma função da aplicação. Seguindo os princípios da modularização, os clientes não tomam conhecimento da maneira pela qual os servidores implementam seus serviços. As entradas e saídas do sistema devem percorrer as sucessivas camadas até alcançarem a aplicação e usuário. Cada camada, à exceção do núcleo funcional e dos dispositivos físicos deve manter um mapeamento da interação nos termos das camadas superiores e inferiores. Os detalhes de interfaceamento são importantes em qualquer implementação deste modelo.

A composição das máquinas abstratas é variada, podendo corresponder aos grupamentos de funcionalidades apresentados no parágrafo anterior: uma aplicação, um servidor do diálogo, um servidor de objetos de interação e um servidor janelas. A progressão no nível de abstração dos serviços prestados em cada máquina pode ser verificada através dos eventos por elas produzidos. Os eventos são gerados por cada máquina em resposta a uma ação do usuário.

Quando o usuário movimenta e aciona o mouse sobre um determinado objeto ocorrem os seguintes eventos. O dispositivo físico transmite um número de incrementos de posição ao controle de dispositivos e em resposta a um acionamento ou liberação, ele interrompe as transmissões sobre um circuito particular. O evento gerado pelo controlador de dispositivos informa ao sistema de janelas a posição atual do mouse e a ocorrência de interrupções. O sistema de janelas informa a seus clientes que um acionamento ocorreu quando o cursor estava em uma determinada posição na tela. Um objeto de interação reconhece as posições e informa a entrada e a saída do mouse de sua área na tela, bem como o acionamento e liberação deste dispositivo durante esta permanência. O controle do diálogo informa ao núcleo funcional que uma determinada funcionalidade está sendo solicitada pelo usuário.

Se existirem múltiplos terminais abstratos ou janelas na tela (cada cliente possui exatamente uma janela no terminal) a tarefa do sistema de janelas torna-se mais complicada. Ele detecta a posição atual do cursor, decide qual cliente está envolvido e realiza um sistema de coordenadas especificamente para a janela deste cliente. O evento é informado ao cliente segundo as coordenadas de sua janela. As solicitações de saídas encaminhadas por um determinado cliente são mapeadas para sua janela específica. Ao contrário do sistema de janelas o controle dos dispositivos desconhece a existência de múltiplos clientes.

Se necessário, a aplicação pode solicitar os serviços diretamente ao controlador dos dispositivos. Sem esta possibilidade de acesso direto, o desempenho do sistema sofreria com os processamentos inúteis da passagem da informação de uma camada intermediária à outra.

3.2.2 - Os modelos de linguagem

Os modelos linguísticos organizam os componentes de um sistema interativo a partir das dimensões semântica, sintática e lexical de uma linguagem interpessoal. No modelo de Seeheim os componentes lógicos de um sistema são agrupados segundo as dimensões apresentação, controle do diálogo e interface com a aplicação.

A apresentação processa os elementos lexicais introduzidos através dos dispositivos físicos para formar unidades sintáticas interpretáveis pelo controle do diálogo. De maneira inversa, ela recebe as expressões liberadas pelo controle do diálogo e as interpreta nos termos das unidades lexicais dos dispositivos de saída. Ela é responsável pela representação física ou imagem do sistema sendo o único componente a manipular os dispositivos físicos e realizar trocas com o usuário. Entretanto ela não se comunica jamais com a aplicação.

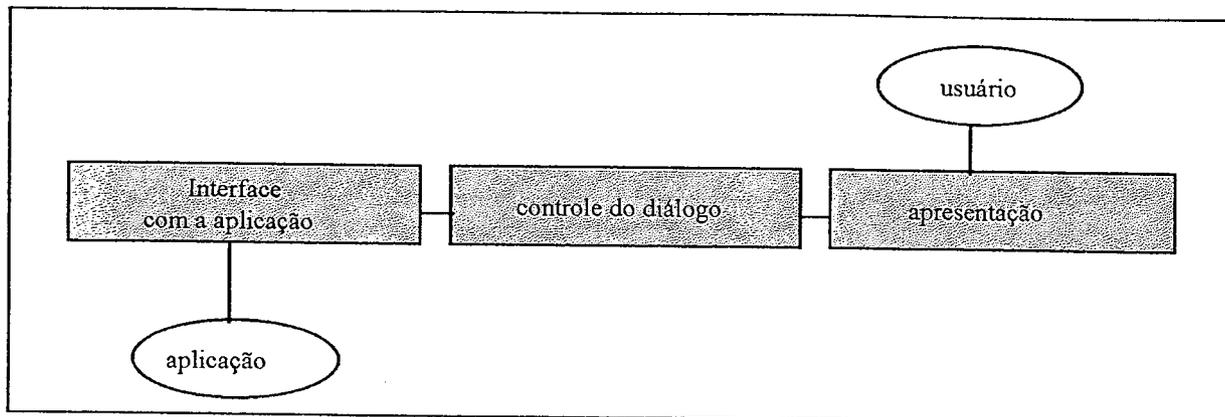


Fig 3.3 - O modelo de Seeheim. (Fonte Coutaz, 1990).

O controle do diálogo desempenha o papel de analisador sintático das unidades significativas provenientes da apresentação. Ele agrupa estas unidades de maneira a formar frases sintaticamente corretas e que correspondem à demandas do usuário à aplicação. Por este modelo cabe ao controle do diálogo gerenciar os estados da interação. Um estado é definido pelo conjunto de demandas possíveis. As funções do controle do diálogo incluem a composição das frases e a manutenção das relações entre os estados.

A interface com a aplicação define um mecanismo de conversão entre as frases montadas pelo controle do diálogo e objetos e ações da aplicação.

O fato do controle do diálogo centralizar o fluxo das transações acarreta uma perda de desempenho quando existem múltiplas "micro" atividades que atraem a atenção do usuário. A contribuição dos modelos multi-agentes diz respeito justamente à descentralização.

3.2.3 - O modelo Multi-agentes

A definição de agente é bastante próxima da de objeto. Agentes são entidades capazes de reagir a estímulos determinados (eventos) produzindo novos estímulos de saída. Um agente é um sistema de tratamento de informações que possui receptores e emissores para adquirir e produzir eventos. Ele possui filtros para selecionar os eventos que interessam e os que não interessam. A memória do agente registra a fila de eventos a serem tratados e seu processador sequencial trata os eventos. Este tratamento ocasiona uma mudança no estado do agente e a produção de um novo evento. Um sistema multi-agentes se caracteriza por uma organização fortemente modular, por tratamentos em paralelo e por comunicações por eventos. Um agente pode ser visto como uma especialização da noção de objeto.

O modelo multi-agentes é essencialmente descentralizador. Ele propõe substituir servidores sequenciais por uma organização de agentes que trabalhem em paralelo. A gestão centralizada do estado da interação é substituída por um repartição de encargos como forma de responder mais eficientemente aos métodos de resolução dos usuários. Os sistemas interativos baseados numa arquitetura multi-agentes proporcionam flexibilidade ao usuário e modularidade ao projetista.

3.2.3.1 - O modelo P.A.C.

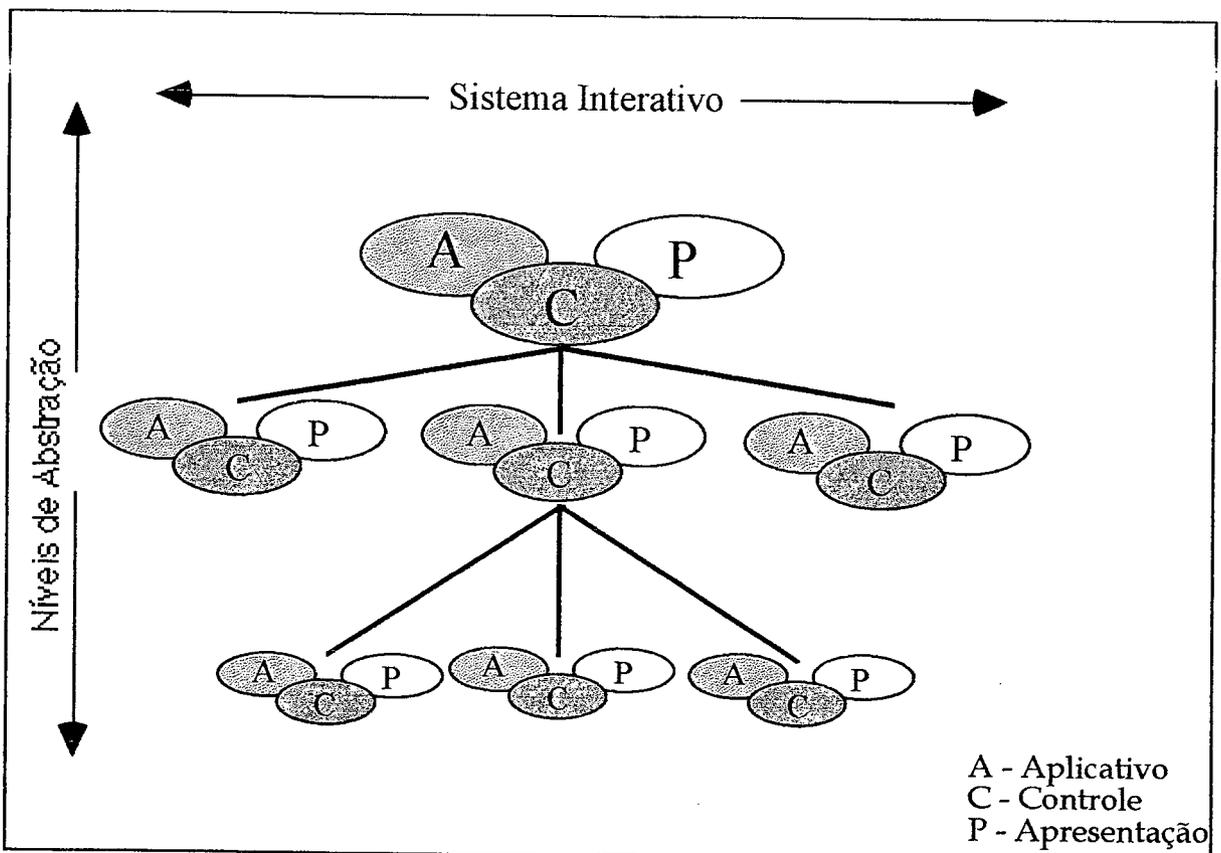


Fig.3.4 - Estrutura de um sistema interativo formada por uma hierarquia de agentes P.A.C. Eles descentralizam a aplicação, o controle e a apresentação através dos diversos níveis de abstração do sistema..

O modelo P.A.C. (*presentation, abstraction et contrôle*) procura explorar as boas características dos modelos já examinados. Assim, ele é essencialmente um modelo multi-agentes que propõe descentralização e paralelismo. A organização de seus componentes resulta das modularizações propostas nos modelos linguísticos e de máquinas abstratas. O modelo PAC descentraliza a função do controle do diálogo distribuindo a distinção entre núcleo funcional e apresentação através dos agentes do sistema. Em outras palavras, cada agente PAC possui uma porção do núcleo funcional, uma porção do controle do diálogo e uma porção da

apresentação do software interativo. Este é finalmente visto como uma hierarquia de agentes PAC.

No topo da hierarquia um sistema interativo é visto como um só agente PAC. Sua aplicação modeliza os conceitos do domínio da tarefa e do gerente do posto de trabalho que são apresentados através de um conjunto de agentes especializados na interação como o usuário. A coerência entre a aplicação e a imagem do sistema é garantida pelo controle de mais alto nível. No nível mais baixo da hierarquia, os objetos PAC elementares definem os parceiros do usuário e são denominados de agentes interativos.

As três dimensões de um agente PAC são: abstração ou aplicação, controle e apresentação. A apresentação manifesta o comportamento do agente. A abstração ou núcleo funcional implementa algum conceito ou competência do domínio de uma maneira desvinculada da apresentação. O controle liga a abstração à apresentação, controla o comportamento das duas dimensões, memoriza o estado atual do agente e mantém os relacionamentos com outros agentes. Um agente interativo PAC é aquele que se relaciona com o usuário.

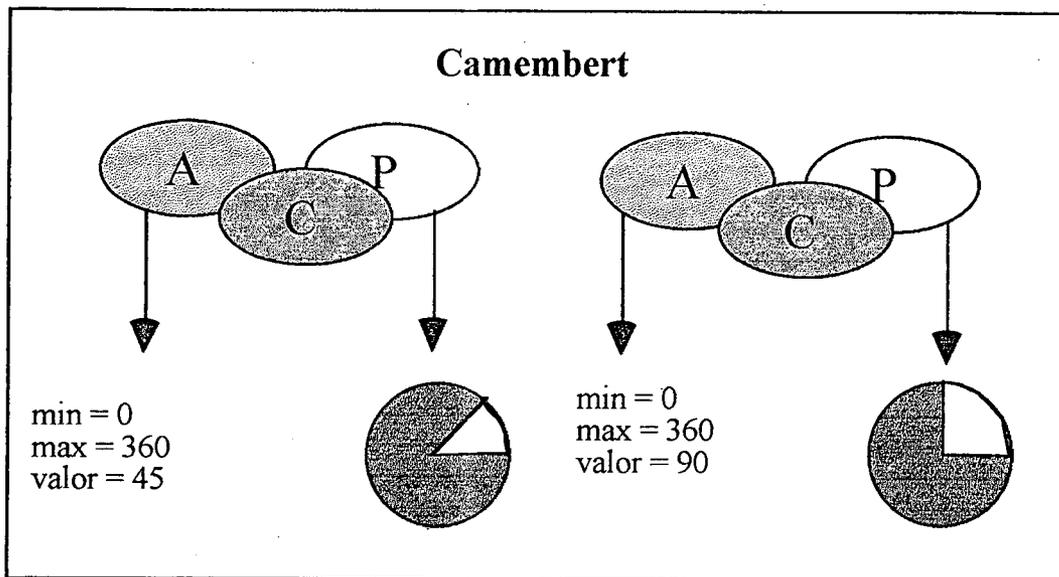


Fig 3.5 - Agente Camembert representando um valor entre 0 e 360.

(Fonte: Coutaz, 1991)

Coutaz apresenta o funcionamento do modelo através do que se poderia chamar, agente Camembert. Trata-se de um agente cuja dimensão aplicação define um valor abstrato dentro de um intervalo de 0 à 360. Este valor é abstrato na medida em que esta dimensão

ignora a maneira através da qual ele é apresentado ao usuário. A dimensão apresentação possui o aspecto que é bastante familiar aos franceses; um queijo Camembert devidamente fatiado. Ela mostra o valor atual ao usuário através de um círculo com dois setores complementares e preenchidos com motivos diferentes. Ela proporciona a entrada de novos valores através da manipulação direta dos setores circulares. Cabe ao controle o papel de manter a coerência entre estas duas dimensões. Assim quando o usuário modifica o tamanho da fatia, a apresentação informa ao controle que solicita à dimensão abstrata que se adapte ao novo valor introduzido. O mesmo ocorre se a atualização for realizada por um outro agente do sistema diretamente sobre o valor abstrato. Neste caso é a dimensão aplicação que informa ao controle a mudança de estado, o qual por sua vez solicita a atualização da apresentação.

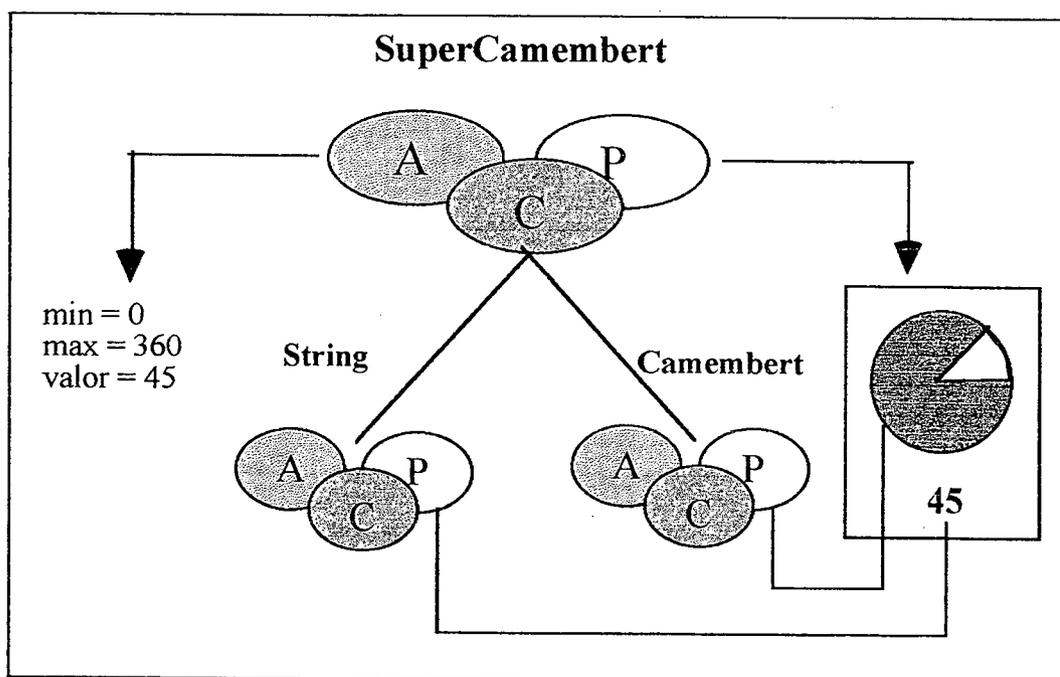


Fig 3.6 - AgenteSuperCamembert formado pelo agente "Camembert" da Fig 3.5 e pelo agente "String". (Fonte: Coutaz, 1991)

Um agente composto é definido a partir de relações entre agentes. Sua aplicação lhe é específica. Sua dimensão apresentação é o somatório das apresentações de todos os agentes. O controle garante a coerência entre abstração e apresentação, mas também gerencia as colaborações e dependências entre os diversos agentes. O agente Super-Camembert mostrado na Fig 3.6 é composto pelos agentes Camembert e pelo agente "String". Este último apresenta um valor abstrato através uma cadeia de caracteres numérica. A apresentação do objeto composto é formada por um quadro e pelas apresentações correspondentes a estes dois agentes

que estão devidamente centradas no quadro. Quando uma ação do sistema modifica o valor abstrato do Super-Camembert, seu controle traduz esta modificação nos termos das abstrações dos agentes que o compõe. Cabe aos controles destes agentes solicitarem a atualização de suas dimensões abstração e apresentação. Uma modificação introduzida pelo usuário pode se dar através da manipulação direta dos setores circulares ou de uma ação de edição sobre a cadeia de caracteres. Ela repercute localmente sobre o agente elementar sendo então transmitida ao controle do agente composto. Este altera o seu valor abstrato e solicita ao controle de seus outros componentes que adaptem suas abstrações e apresentações ao novo valor.

O controle é constituído por protocolos de comunicação que envolvem os métodos das dimensões abstração e da apresentação. Ele atua como árbitro e tradutor. Assim, além de controlar a coerência entre estas duas dimensões ele é o intermediário entre os formalismos naturalmente distintos de uma e de outra.

A modularidade do sistema fica garantida pois abstração e apresentação não se comunicam diretamente. Este fato facilita a substituição dos objetos interativos que fazem a apresentação de um mesmo conceito. A flexibilidade na interação é garantida pela possibilidade de diálogos distribuídos entre diferentes agentes. Devido às suas três memórias de estados: abstrato, controle e apresentação, o agente pode ser abandonado e retomado pelo usuário a qualquer instante. Assim o usuário tem a facilidade de escolher com qual agente interagir. As apresentações de diversos conceitos podem ser manifestadas simultaneamente e da mesma forma o usuário pode agir simultaneamente sobre elas.

A semântica e a sintaxe estão repartidas entre os agentes do sistema. Mesmo os agentes interativos elementares que se encontram no nível mais baixo da hierarquia possuem estas duas componentes. Esta repartição semântica facilita a extensão e a reparação da aplicação e reduz as trocas entre agentes.

3.3 - As ferramentas para a construção de interfaces

As ferramentas para a construção de interfaces podem ser classificadas segundo as categorias: toolboxes, esqueletos de aplicação e geradores de interfaces. As toolboxes definem uma biblioteca de objetos de interação. Os esqueletos de aplicação definem a porção do diálogo que é reutilizável.

3.3.1 - Toolboxes

Uma toolbox corresponde a uma biblioteca de objetos de interação que é colocada à disposição dos projetistas por uma plataforma de desenvolvimento específica. Ela deve proporcionar consistência, abrangência em termos de objetos de interação, facilidades para especializações e um bom desempenho geral. Estas ferramentas impõem estilos de apresentação e de comportamento, que em geral levam a uma maior consistência nos projetos. No entanto elas podem impor algumas restrições caso não possuam um conjunto de objetos suficientemente abrangente e versátil. A primeira solução para o problema de falta de abrangência de uma toolkit implica na especialização de novos objetos pelo projetista, o que pode não ser, dependendo da toolkit, uma tarefa fácil. Isto depende de uma verdadeira orientação a objetos. Outra solução seria a combinação de diferentes toolkits para obter o conjunto adequado de objetos. Esta solução implica entretanto, na existência de diferentes estilos de objetos e em problemas de desempenho geral. Outros indicativos na escolha de uma toolkit incluem sua disseminação como produto, sua portabilidade em relação aos sistemas de janelas e do apoio do fabricante.

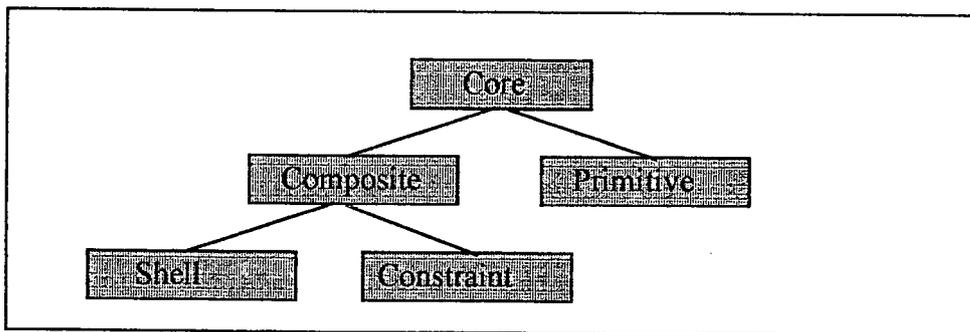


Fig. 4.2 - Arquitetura da biblioteca de objetos X Toolkit Intrinsic.

Os widgets, ou objetos de interação da biblioteca X Toolkit Intrinsic (ambiente X Window), são implementados como janelas. Assim a meta-classe Core, raiz da arquitetura de X Toolkit, apresenta atributos sobre a localização e o tamanho do widget e métodos para criá-lo e manipulá-lo na tela. Abaixo de Core a arquitetura distingue classes de primitivas e de objetos compostos. As classes de objetos compostos dividem-se entre aquelas que gerenciam relações entre seus componentes e aquelas que fornecem uma interface com o gerente de janelas (shell).

A denominação em termos de pais e filhos (parents-children) para designar o objeto composto e seus componentes, pode gerar alguma confusão com a relação de herança. A

hierarquia verificada entre os componentes de um objeto composto não está baseada em especializações, mas na composição.

As classes de objetos compostos são definidas a partir das relações verificadas entre seus componentes. Assim em uma composição simples o objeto composto do tipo "container" assegura a coerência entre as posições de seus componentes. Ao ser apresentado ou movimentado na tela o container apresenta e movimenta todos os seus componentes conjuntamente. Este serviço proporciona aos clientes deste tipo de objeto uma economia de especificação considerável, na medida em que eles não precisam especificar as posições de todos os objetos. Exemplos de objetos construídos como containers são os menus, caixas de diálogo, janelas, etc.

Uma especialização de container gerencia as relações geométricas (dimensões) de seus componentes. Ao ser apresentado o objeto composto posiciona e redimensiona automaticamente seus componentes para que caibam em sua área. Uma composição do tipo "restrição" ou "constraint" impõe o relacionamento entre objetos através de uma equação ou um sistema de equações. Ela pode ser utilizada para associar o valor de uma variável do núcleo funcional com uma característica da apresentação um objeto de interação.

O desafio ligado à evolução de toolkits envolve a definição de novas classes de objetos mais adequadas e mais ricas do ponto de vista da interação entre eles, com usuários e com os projetistas.

3.3.2 - Esqueletos de aplicação

Os chamados esqueletos de aplicação permitem a montagem de um programa reutilizável e extensível para a concepção de interfaces. Eles apresentam módulos que asseguram grande parte dos serviços gerais de interação com o usuário, como funções de ajuda e tutoriais, gestão de erros, facilidades de anular-refazer, cortar-colar, etc.... A fig. 3.7 mostra um esqueleto de aplicação baseado no modelo de arquitetura PAC. Assim o controle reutilizável desempenha o papel de árbitro e de tradutor entre serviços de dois níveis de abstração. Os mecanismos de indireção que permitem esta tradução se baseiam em regras de produção, tabelas de ligação, base de dados ativas e passivas. APEX (Application Extensible) é um esqueleto de aplicação que segue o modelo de arquitetura PAC.

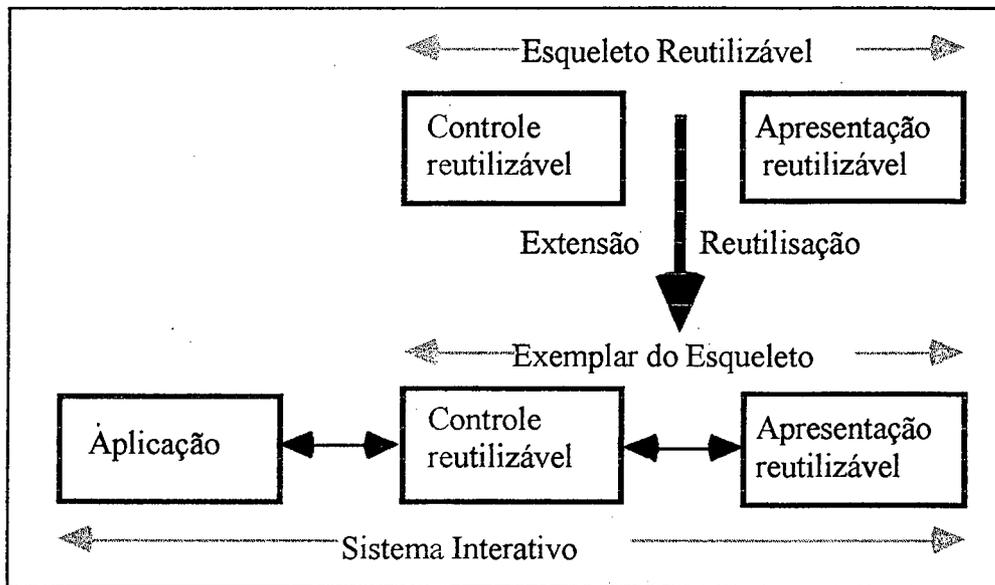


Fig 3.7 - Esqueleto de aplicação baseado no modelo PAC (Fonte: Coutaz, 1991)

3.3.3 - Geradores de interface

Um sistema gerador de interfaces produz automaticamente a interface de um sistema interativo a partir de uma especificação. Os sistemas se diferenciam em função de suas complexidades. Alguns geram somente a apresentação enquanto que outros são capazes de gerar também o controle do diálogo.

O sistema Trident (Bodart & Vanderdonckt, 1993a) é um gerador automático que seleciona e distribui os objetos de interação a partir descrição da semântica da aplicação. Esta descrição se baseia na estrutura e no fluxo dos dados durante a realização da tarefa. A estrutura de dados é descrita através de um formalismo entidade-relacionamento-atributo e o fluxo dos dados, por um gráfico do encadeamento de funções. As regras de seleção de objetos são derivadas de diversos guias de estilo e de recomendações ergonômicas de diferentes autores.

Estas regras relacionam quatro tipos de informações para a seleção de um objeto. (i) A primeira informação envolve um tipo do dado sendo manipulado em entrada ou saída. (ii) A segunda informação processada na regra envolve cinco metadados, definidos como informações adicionais sobre a utilização do dado. (iii) O terceiro parâmetro especifica as restrições de espaço na tela e o (iv) último modela o usuário segundo suas experiências e preferências.

Os objetos de interação deste sistema são chamados de objetos de interação abstratos - OIA -. Seus autores definem um OIA como um interador lógico baseado tão somente nos aspectos de comportamento dos objetos de interação concretos encontrados nas diferentes toolboxes. Eles definem as classes de OIA apresentadas na Tabela 3.1.

| Classes de OIA | Objetos de Interação Abstratos |
|----------------|--|
| Ação | menu, item de menu, barra de menu, menu drop-down, menu em cascata, sub-menu, etc... |
| Rolamento | flexa, cursor, barra de rolamento, quadro |
| Estático | rótulo, separador, grupamento, prompt, ícone |
| Controle | edition box, escala, dial, check box, interruptor, radio box, push button, list box, combination box, tabela, etc. |
| Diálogo | janela, janela de ajuda, caixa de diálogo, painel, etc. |
| Feedback | mensagem, indicador de progressão, cursor contextual, etc. |

Tabela 3.1 - Tipologia dos OIA. (Fonte: Vanderdonckt e Bodart, 1993)

Após terem sido selecionados os AIOs são mapeados sobre um determinado ambiente e transformados em objetos de interação concretos. Estes últimos são então arranjados na tela conforme uma hierarquia de objetos derivada da descrição da estrutura de dados.

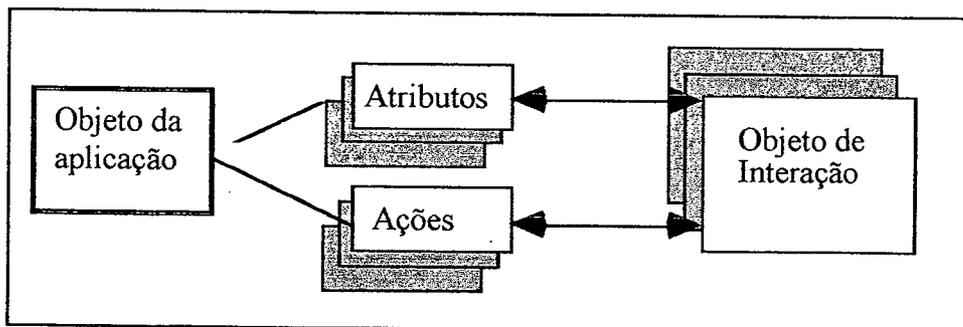


Fig 3.8 - Relação entre objeto da aplicação e objeto de interação. (Fonte: Baar, Foley & Mullet, 1992)

A abordagem apresentada por Baar, Foley e Mullet (Baar, Foley, & Mullet, 1992) para gerar automaticamente uma interface gráfica com o usuário parte da descrição do modelo de

dados da aplicação feita através de um editor gráfico especializado (D2M2edit). As ações e atributos dos objetos deste modelo são associadas através de regras de produção com os componentes da arquitetura Open Look disponíveis na ferramenta de projeto Devguide. O motor de inferência apresenta regras para a seleção e para a organização dos componentes que são extraídas do guia de estilo Open Look.

Um objeto de dado é descrito através de informações sobre suas características específicas e de uso (metadado). O metadado é fator essencial para a seleção exitosa de um objeto de interação. O motor de inferência é projetado para ser independente de uma toolkit específica. Suas conclusões se referem especificamente a componentes da arquitetura Open Look, mas eles podem ser realizados através de widgets de diversos fabricantes.

A toolkit de ViewPoint (Johnson, Roberts, Verplank, Smith, Irby, Beard, et al., 1989) apresenta widgets que buscam uma aproximação com a semântica do valor a ser apresentado ou entrado pelo usuário. Os exemplos interessantes são os interadores; comandos, booleanos, escolha simples, múltipla escolha, escolha com menu, pop-up menu, texto e texto com menu.

No sistema Jade (Zanden & Myers, 1990) o programador especifica a aplicação através de uma linguagem declarativa onde ele escolhe entre 7 tipos de widgets, independentes dos aspectos da apresentação. Estas categorias são: escolha simples, múltipla escolha, texto, escolha simples com texto, múltipla escolha com texto (estes interadores permitem que o usuário entre uma cadeia de caracteres que vai determinar a escolha), comando (corresponde à um item dentro de um menu), número-num-intervalo (permite a entrada de um valor dentro de um intervalo especificado). A apresentação destes interadores é implementada através do processamento de uma "folha de estilo" onde o projetista gráfico define as aparências dos interadores e as regras para o seu posicionamento na tela. Podem existir diversas folhas de estilo, por exemplo uma para cada ambiente de concepção. A definição do sistema resulta da compilação da especificação da aplicação e da folha de estilo.

O sistema Its (Gould, Boies, & Lewis, 1991) tem as mesmas características de Jade, mas propõe mecanismos mais refinados para a definição da semântica dos valores da aplicação e de suas correspondências com uma apresentação. Assim uma semântica do tipo um-entre-n-possibilidades pode ser apresentada como um conjunto de botões radio (radio-button) se ela se refere ao domínio da tarefa (objeto semântico) ou por um menu pull-down, se ela se refere à um parâmetro de controle do sistema computacional (objeto sintático).

A abordagem do sistema Ace (Johnson, 1992) se baseia na existência de semânticas reutilizáveis para definir classes de objetos semânticos. As semânticas reutilizáveis oferecidas por Ace correspondem a: valores de dados, apresentação de dados, fazer escolhas e apresentação de escolhas. ACE propõe duas classes de "Selectors"; "data-selectors" e "command-selectors". Os tipos dos seletores de dados são: um-dentro-de-um-conjunto-pequeno-e-discreto, diversos-dentro-de-um-conjunto-discreto, um-dentro-de-um-conjuto-grande-e-contínuo, diversos-dentro-de-um-conjunto-grande-e-ordenado. O primeiro tipo de seletor por exemplo, pode ser apresentado por botões rádio, campos de edição ou menus. Os diversos tipos de "Presenters" definem a aparência dos widgets, de seu layout e de seu comportamento imediato.

3.4 - Conclusões

Como componente funcional um OIA^e corresponde à noção de objeto de interação, que realiza os aspectos estáticos de uma interface homem-computador e se comunica com o usuário. Embora ele participe do diálogo ele não determina a sequência das interações.

As arquiteturas de sistemas interativos seguem modelos linguísticos que colocam em módulos distintos os serviços do núcleo funcional e da apresentação, controlados pelo controle do diálogo. Isto de maneira mais ou menos centralizada. Sob o ponto de vista das arquiteturas de sistemas interativos, um OIA^e adere naturalmente ao modelo PAC. Sua morfologia é a de um agente interativo PAC, com atributos de abstração, controle e apresentação.

A lógica dos geradores de sistemas interativos está baseada no trabalho prescrito, o que se reflete na especificação do núcleo funcional segundo a estrutura e o fluxo de dados. Na proposta do ambiente Archie a especificação do núcleo funcional prevê também a descrição das atividades realizadas, segundo o formalismo MAD.

A noção de OIA^e representa uma evolução em relação aos objetos de interação abstratos propostos nos diversos sistemas examinados. Seu caráter inédito está ligado à abstração ergonômica. Um OIA^e facilita o raciocínio baseado nas regras de produção, pois se estabelece uma correspondência efetiva entre seus atributos e as recomendações ergonômicas que estão na sua origem.

Os OIA^e diferenciam-se dos objetos de interação convencionais, na medida em que eles estão ligados à lógica de utilização do sistema, que é a lógica tratada nos textos das

recomendações ergonômicas. Aí as questões principais estão ligadas ao significado e às inter-relações entre sinais

O capítulo seguinte é dedicado à perspectiva semiótica de um interface homem-computador.

4 - IHC - PERSPECTIVA SEMIÓTICA

A apresentação do arcabouço teórico da semiótica em sua aplicação ao projeto de sistemas interativos propõe o redirecionamento na multidisciplinariedade desta questão. O computador é visto como uma máquina (simbólica) que usa os sinais introduzidos pelos programadores para produzir os sinais que os usuários manipulam.

Os sinais possuem componentes que estimulam tanto os mecanismos sensoriais do cognitivo humano como os mecanismos convencionados pelas interações sociais. Muitas das contribuições trazidas pelos psicólogos são inapropriadas para responder questões sobre a interação homem-computador que estão culturalmente determinadas. Nestas situações uma abordagem antropológica, com observações detalhadas e uma imersão na cultura deve ser considerada.

Os AIO^e são definidos sinais como forma de alertar projetistas e programadores sobre as implicações antropológicas de sua tarefa. Assim como é importante que a forma da expressão dos sinais favoreça sua percepção e seja compatível com seu conteúdo é igualmente importante que ela esteja inserida na convenção social em que o sistema deve operar.

Este capítulo propõe uma apresentação geral da ciência dos signos e de seus pressupostos básicos; as semiósis. A maior parte dele é dedicada a apresentação da "Teoria da Semiótica Computacional" (Andersen, 1990) e de sua tipologia de sinais computacionais.

4.1 - A semiótica

" A civilização humana depende dos sinais e dos sistemas de sinais; a inteligência humana é inseparável do funcionamento dos sinais."

Morris, C. -Fundamentals of the theory of signs.

A semiótica é a ciência que estuda os sistemas de sinais: linguagens, códigos, sinalização, etc. Devido ao papel privilegiado que assumem os sistemas de linguagem muitos autores consideram a semiótica como a ciência dos sinais não linguísticos. A visão que se pretende passar é da semiótica como a ciência mãe, da qual a linguística herda as características principais. Segundo Guiraud (Guiraud, 1983), a origem desta ciência data do início do século

XX, época em que dois autores, Saussure e Pierce, lançaram as bases da teoria geral dos sinais, chamada por um de Semiologia e por outro de Semiótica. Saussure salienta a função social do sinais enquanto Pierce realça sua função lógica. Ainda segundo Guiraud o termo "semiologia" é empregado na Europa e "semiótica" nos EUA.

4.1.1 - Sinais e esquemas semióticos

Existem inúmeras definições de sinal e inúmeros esquemas representando seus componentes e suas interrelações (semiósisis). A terminologia neles empregada se assemelha, porém os significados dos termos varia de modelo para modelo. Nos próximos parágrafos serão apresentados os esquemas semióticos considerados no desenvolvimento deste trabalho.

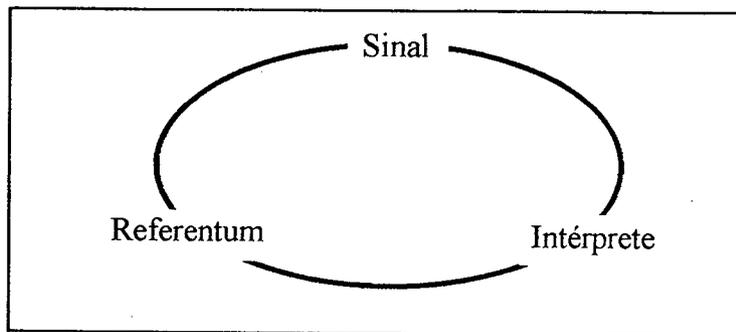


Fig. 4.1 - Esquema semiótico de Pierce

Morris (Morris, 1974) apresenta a tríade de Pierce envolvendo sinal **S**, referentum **R** e intérprete **I**. Assim, **S** é um sinal de **R** para **I** na medida em que **I** toma conhecimento de **R** a partir da presença de **S**.

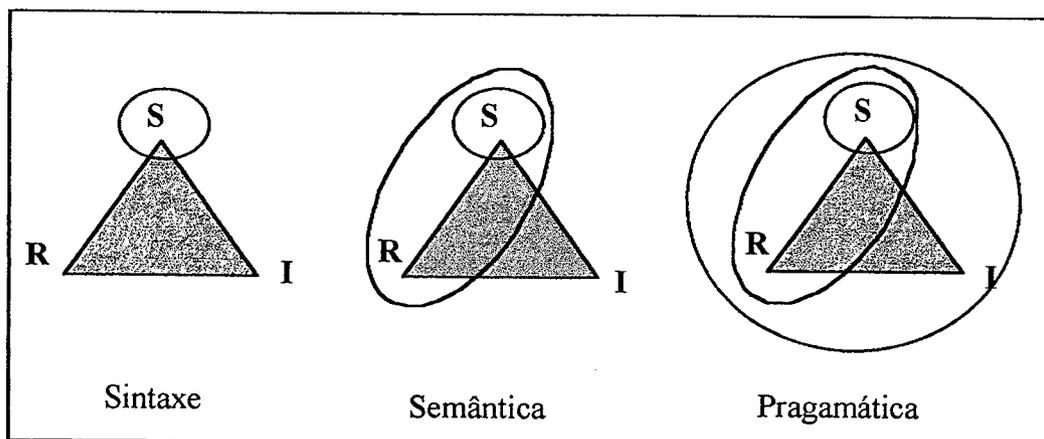


Fig 4.2 - As relações semióticas (Fonte: Nadin, 1988)

Segundo este esquema um sinal ocorre somente quando ele for interpretado por um interpretador. As relações envolvendo os três fatores de um sinal definem as dimensões pragmática, semântica e sintática da semiótica.

A relação pragmática relaciona os sinais com seus intérpretes e é descrita tipicamente através do termo "exprime". A semântica, descrita em termos de "designa" ou "denota", é a relação envolvendo de uma lado os sinais e por outro os objetos que eles representam. A relação entre dois ou mais sinais de um sistema define sua sintaxe e é descrita em termos de "determina", "pressupõe", etc.

A partir destes níveis semióticos pode-se derivar as três funções fundamentais de um sinal (Nadin, 1988). Um sinal é visto como expressão quando a ênfase está colocada sobre suas qualidades de apresentação. Ele é visto como representação quando além destas qualidades a ênfase estiver na relação da representação com o objeto representado. Ele é visto como conhecimento quando o interpretador integrar as funções de expressão e de representação e conseguir derivar algum conhecimento geral ou específico sobre o objeto representado.

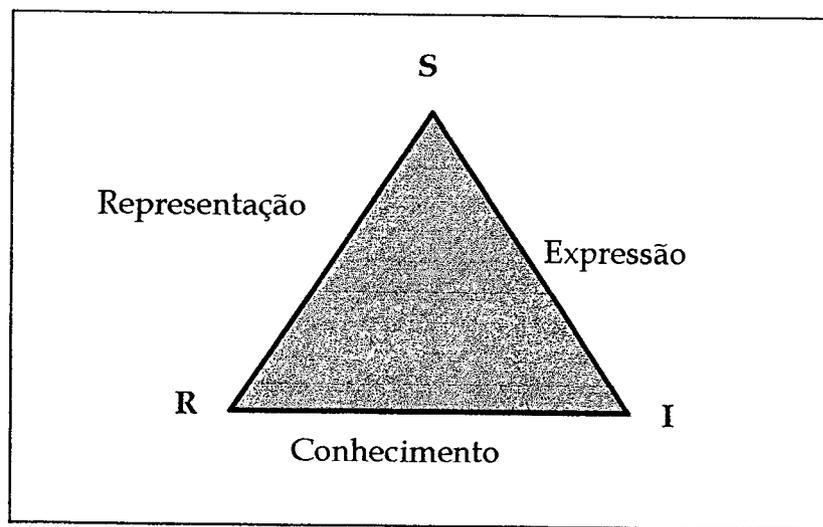


Fig 4.3 - As funções de um sinal

Guiraud define o sinal como um estímulo, isto é uma substância sensível, cuja imagem mental é associada em nosso cognitivo com a de um outro estímulo. Ele desempenha o papel de evocador da comunicação. O esquema semiótico no qual este autor se baseia é o da teoria da comunicação e seus personagens são o emissor, a mensagem, a referência, o código e o receptor. Ele cita Jakobson (pag 9) ao apresentar as funções semióticas a partir das relações

entre estes personagens. As principais envolvem mensagem-referência e emissor-mensagem. A primeira define uma função lógica e objetiva que visa evitar toda e qualquer confusão entre o sinal e o objeto representado. A segunda é uma função emotiva e subjetiva. Uma mensagem, além de seu conteúdo objetivo, carrega também a atitude do emissor frente ao objeto. Relações objetivas e afetivas, são as bases, ao mesmo tempo complementares e concorrentes da comunicação.

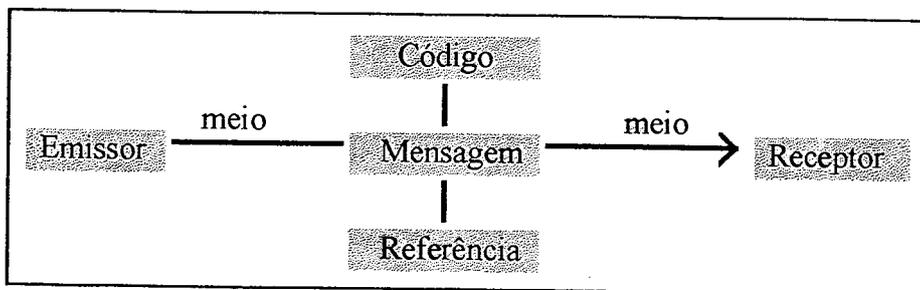


Fig 4.4 - Esquema semiótico baseado na teoria da comunicação

Um código define a convenção entre significado e significante. Ele resulta de um acordo entre os usuários de um sistema de sinais que reconhecem esta relação e a respeitam no emprego do sinal. Este acordo pode ser mais ou menos explícito, o que separa dois grandes tipos de relações: as motivadas (implícitas) e as arbitrarias (explícitas). As motivações se verificam quando existe uma relação natural entre mensagem e referência. É o caso das analogias que emprestam aos símbolos e ícones (imagens), de um modo mais ou menos abstrato, a aparência dos objetos ou das funcionalidades que eles representam. Nos formalismos das ciências exatas os códigos são geralmente arbitrários e funcionam por pura convenção. Além disto sua eficácia é garantida por uma correspondência unívoca entre mensagem e referência (monosemia). Nos chamados códigos estéticos ou poéticos verifica-se em geral uma convenção enfraquecida por uma polisemia uma expressão ligada a diversos conteúdos, cabendo ao receptor escolher um sentido entre os diversos possíveis. A ambiguidade do sinal polisêmico geralmente desaparece quando se considera o contexto da mensagem.

Guiraud propõe uma tipologia de códigos lógicos baseada em seus objetivos. Assim, eles são divididos segundo os paralinguísticos, os práticos, os epistemológicos e os mânticos. Os paralinguísticos tem a função de substituir a linguagem articulada em caso de algum tipo de restrição e incluem códigos como o morse, o braile, de bandeiras, de mãos (surdo-mudos). Os códigos práticos tem por objetivo coordenar ações na circulação em geral, na programação da

produção, na operação de sistemas, etc. Seus sinais são injunções, instruções, avisos, e alarmes. Os códigos epistemológicos tem o objetivo de informar sobre a identidade dos elementos e de suas relações em um determinado campo do conhecimento. Os códigos mânticos dizem respeito às artes da adivinhação e da comunicação com os deuses e incluem a astrologia, a cartomancia, quiromancia, etc.

Os códigos estéticos se referem aos sentimentos íntimos e incluem principalmente as artes e literaturas. Seus sinais são icônicos e analógicos. Uma terceira categoria de códigos, os sociais, são fruto de uma dupla experiência, ao mesmo tempo objetiva e subjetiva, do homem em sociedade. Neste tipo de código existe parcialidade, o emissor tem uma participação direta no fato que acaba de relatar. Assim seus sinais identificam a situação dos indivíduos frente ao grupo e do grupo frente a coletividade. Eles são insígnias, armas, uniformes, tatuagens, nomes e sobrenomes, maneiras, etc. Seus códigos incluem protocolos, rituais, jogos, e modas.

É importante citar Ware (Ware, 1992) que divide os sinais segundo representações sensoriais e convencionadas. Os sinais convencionados subentendem o esquema semiótico apresentado acima. Já os sensoriais estão ligados às primeiras etapas do processamento sensorial da informação. Eles tendem a ser estáveis frente a indivíduos e culturas. Ele cita os trabalhos de Bertin (pag. 2) que descrevem a maneira através das quais marcações gráficas transmitem significado baseado em julgamentos subjetivos. Os elementos básicos da gramática sensorial estão baseados em estruturas fisicamente presentes no mundo. As leis da Gestalt, derivadas dos próprios mecanismos da percepção de objetos, fornecem exemplos de sinais cujo significado é definido nas primeiras etapas da cognição. Os fatos dos objetos possuem superfícies, estarem sujeitos à lei da gravidade e da luz se propagar em linha reta independem de uma cultura específica. O interesse na identificação de uma gramática sensorial envolve a naturalidade e a facilidade de utilização de um esquema representacional que seja válido em uma grande variedade de contextos.

Prieto (Prieto, 1972) define um sinal como um instrumento para transmitir mensagens. Por significado este autor denota uma classe formada por todas as mensagens que um sinal admite. Inversamente, por significante entende-se a classe formada por todas os sinais que uma mensagem admite. Mensagem e sinal são instâncias de significado e significante. O procedimento de análise denominado "teste de comutação" permite a identificação destas duas classes de variantes. Prieto apresenta os mecanismos de indicação que caracterizam um ato

simbólico e que são de dois tipos: indicação notificativa e indicação significativa. A primeira indicação é fornecida pelo sinal ao ser produzido e simplesmente indica ao receptor que o emissor se propõe a emitir um sinal. A segunda indicação informa ao receptor que a classe à qual pertence a mensagem que chega é uma classe familiar, isto é, capaz de ser tratada. A operação final consiste na seleção de uma entre todas as mensagens que compõem a classe de significados para a sua interpretação.

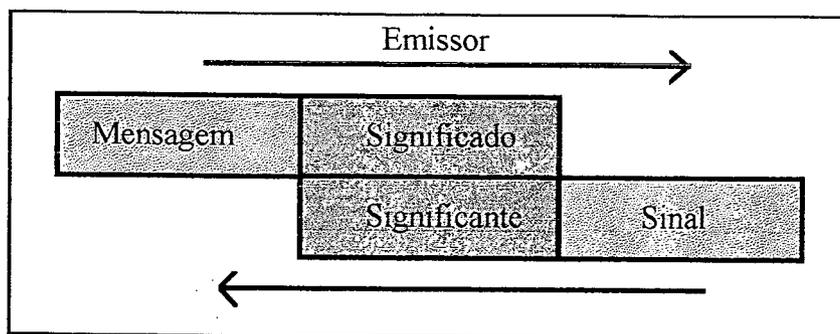


Fig 4.5 - Esquema semiótico proposto por Prieto

Hjelmslev (citado por Andersen, 1990) define sinal como uma relação ou uma função que associa um conteúdo a uma expressão. A expressão é a dimensão manifesta de um ato simbólico. Ela pode envolver diversas substâncias, por exemplo: gestos, movimentos, sons, pontos no papel, pixels na tela, etc. O conteúdo de um sinal se realiza na mente da pessoa que o interpreta e corresponde a um conhecimento sobre um objeto ou propriedade do mundo. As dimensões conteúdo e expressão são interdependentes, o que significa que um sinal não existe sem uma delas.

Conteúdo e expressão apresentam contínuo, substância e forma. A substância representa uma característica do contínuo que é instanciada por uma forma. A forma surge no momento do ato simbólico quando a substância instanciada passa a ser diferenciável em relação à uma outra instância e pertinente em relação ao conteúdo ou à expressão. Desta forma, os fatores decisivos em um sinal são suas formas.

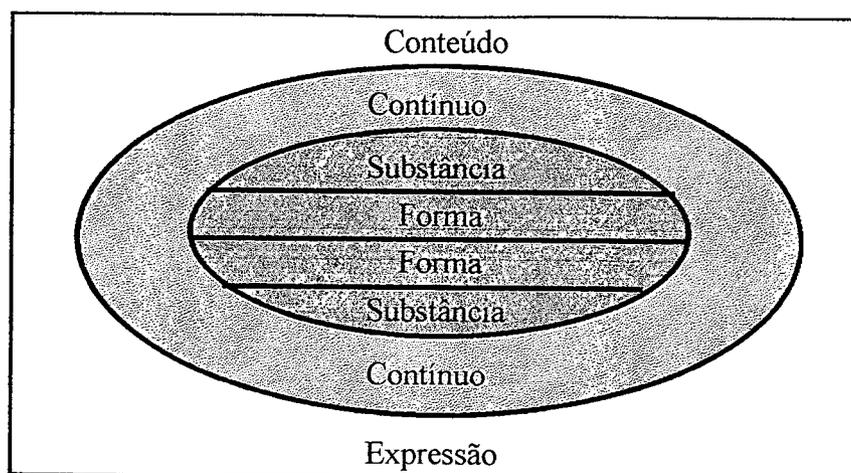


Fig 4.6 - Esquema semiótico baseado no estruturalismo

As formas de sinais compostos podem ser articuladas. Existem dois tipos de articulações: a primeira e a segunda. A primeira articulação se verifica quando existe uma correspondência simbólica entre os fatores da forma da expressão e os fatores da forma do conteúdo de um sinal. Estes fatores são denominados signos. Por exemplo, o número 201 indica um apartamento localizado no segundo andar (2), de frente e à direita da fachada (01). A segunda articulação se verifica em um sinal já articulado cujos signos não são formados por outros signos. Neste caso estes fatores são denominados de figuras. As figuras ocorrem quando não existe uma correspondência entre os fatores da forma de expressão e de conteúdo de um sinal ou de um signo. Os sinais da linguagem escrita ou falada apresentam dupla articulação, na medida em que grafemas ou fonemas, os componentes elementares das palavras, constituem figuras.

Um terceiro fator a participar na formação de um ato simbólico consiste na distinção entre os aspectos de seu sistema e de seu processo. A descrição do sistema envolve funções entre entidades que concorrem pela mesma posição no ato simbólico. A descrição do processo ou do uso semiótico envolve as relações entre entidades que ocorrem juntas no mesmo ato simbólico. As relações de um processo são descritas por sintagmas e as de um sistema por paradigmas. Um dos pontos básicos do estruturalismo reside no fato de que as unidades de um sistema são definidas de uma maneira relacional. Deste modo elas não são definidas em função da matéria de que são feitas, mas pelas relações com as outras entidades do sistema. Os três tipos de relações ou funções são: interdependência, determinação e constelação. Os

participantes das funções são chamados functivos, que podem ser de dois tipos: constantes e variáveis.

A interdependência é uma função com duas constantes. Toda vez que "a" ocorre, "b" também ocorre e vice e versa ($a \leftrightarrow b$). A determinação ou dependência envolve uma constante e uma variável, por exemplo "a" determina a ocorrência de "b" ($a \rightarrow b$). A constelação é uma função que envolve somente variáveis "a" e "b" ocorrem livremente ($a \dashv b$).

O fator determinante na definição dos componentes de um sistema semiótico são as coesões entre as partes resultantes de suas divisões. As coesões são funções que apresentam no mínimo uma constante, como em uma interdependência ou em uma determinação. O procedimento analítico consiste em dividir o ato simbólico em partes significativas que estabelecem coesões entre si. Assim sucessivamente até o momento em que as partes resultantes não sejam significativas. Na análise de um texto por exemplo, a primeira coesão que surge se refere à determinação de uma sentença subordinada por uma sentença principal. As partes, ou as sentenças são chamadas de lexemas e o todo ou o período, de léxias. A análise prossegue em busca de coesões dentro da sentença até chegar nas unidades elementares chamadas taxemas que correspondem aos radicais e terminações que formam as palavras.

4.2 - A semiótica computacional

Andersen (Andersen, 1990) propõe a semiótica computacional como disciplina que estuda a natureza e a utilização de sinais computacionais na sociedade atual. Suas propostas são baseadas em duas das interpretações do esquema semiótico apresentadas no tópico anterior: o esquema estruturalista elaborado por Hjelmslev e a tríade de Pierce. No centro de sua perspectiva está o indivíduo, considerado como o criador, o intérprete e a referência dos sinais. Ele usa a produção semiótica de outros para (re)produzir conhecimento comum. Um sinal é uma relação entre formas de expressão e de conteúdo que só ocorre quando ele é interpretado. Assim, não se pode dizer que um projetista conceba sinais. Ele propõe sinais, que em algumas circunstâncias se realizam, mas que em muitas outras nunca atingem a realização. Programar, no sentido semiótico do termo é, segundo Andersen (Andersen, 1993) usar o computador para tentar dizer algo às pessoas.

Este autor vê o computador como um meio para a comunicação. Ele se baseia na teoria de comunicação para argumentar que este processo pressupõe atos simbólicos articulados por um emissor, segundo um sistema simbólico, e endereçado a um receptor. Em um sistema informatizado o projetista atua como emissor ou receptor. Ele define os limites da comunicação criando os sinais que o usuário pode manipular. Segundo Andersen o computador não possui as faculdades de um emissor ou de um receptor. As pessoas articulam uma linguagem mesmo sem conhecer seu "programa" ou gramática e possuem ainda a capacidade de modificá-lo naturalmente. As linguagens humanas não foram construídas por um grupo de projetistas, mas evoluíram naturalmente com o uso.

O mapeamento da semiótica computacional apresenta três perspectivas; a psicológica, a social e a artística (sinal como conhecimento, sinal como estímulo e sinal como artefato). Na perspectiva psicológica o sinal é visto como conhecimento, e o indivíduo é considerado como um conjunto de estruturas e de mecanismos capaz de entender e de aprender a usar os sinais. Na perspectiva social ou comportamental o indivíduo é considerado como uma entidade indivisível sendo o sinal tratado como estímulo em suas interações com o ambiente, aí incluída a comunicação com outros indivíduos. Na visão de sinal como artefato o indivíduo é considerado como explorador e inovador na criação de significados.

O conteúdo de um sistema informatizado está no sistema de referência. Este sistema é definido como parte do mundo que é escolhida para ser vista como um todo composto de estrutura e processo. O sistema informatizado é a expressão do sistema de referência. Supondo-se que ambos os sistemas estejam estruturados da mesma forma, pode-se então dizer que os planos de conteúdo e de expressão de um sistema informatizado são homomórficos. Ele é assim visto como um sistema de expressões, até um certo ponto vazias, pois dependem do usuário para se realizarem como sinais. Os projetistas podem influenciar fortemente estas interpretações ao conceberem seus candidatos a sinais computacionais. Assim, sua atividade possui o caráter de criação de significados.

Para estudar um sistema computacional Andersen estendeu e adaptou as funções básicas da teoria estruturalista de maneira a cobrir atos simbólicos e atos não simbólicos em suas interações. O objetivo é de poder entender a relação dos sinais com o contexto do trabalho. Assim pode-se dividir um processo não simbólico em partes cada vez menores que são definidas a partir das funções que elas estabelecem entre si. Estas funções correspondem

aos sintagmas e paradigmas de ação. Um sintagma de ação refere-se à sequência de entrada de parâmetros. O sintagma geral "objeto+comando", utilizado nas interfaces gráficas interativas, tem a vantagem de permitir ao projetista prevenir erros de interação pela remoção de ações ilegais após a seleção do objeto. Um paradigma de ação especifica o conjunto de ações disponíveis em um determinado momento, o paradigma proposto pelo sistema deve ser compatível com as necessidades do usuário. O objeto de interface chamado "menu" pode ser interpretado como uma manifestação deste paradigma. Um sistema possui estrutura e processo. A estrutura delimita os caminhos do processo, entendido como as transformações de uma parte do mundo durante um determinado período de tempo.

4.2.1 - Sinais Computacionais

Os sinais computacionais são definidos como aqueles cujo plano de expressão se manifesta no processo de mudança da substância dos dispositivos de entrada e de saída do sistema informatizado. Seu conteúdo está no sistema de referência. A interface homem-computador é vista como uma coleção de sinais computacionais, isto é, toda a parte do processo do sistema que é detectada, utilizada e interpretada por uma comunidade de usuários. Ela deixa de ser vista como componente e passa a ser entendida como processo de um sistema. Segundo esta definição pode-se afirmar que um sistema informatizado possui inúmeras interfaces, uma vez que cada usuário entra em contato com uma coleção diferente de sinais os quais ele interpreta de uma maneira particular. A relação que se estabelece entre o usuário e as partes perceptíveis do sistema faz com que uma nova interface emerja do sistema informatizado cada vez ele é utilizado.

Os sinais computacionais formam estruturas de propriedades manipuláveis, permanentes e transitórias que podem realizar ações sobre os outros sinais do sistema. As propriedades manipuláveis são produzidas pelo usuário com o objetivo de articular suas ações e incluem o pressionar de uma tecla, os movimentos do "mouse", etc. As propriedades permanentes, geradas pelo computador, são aquelas que permanecem constantes durante o ciclo de vida ativa do sinal e que servem para diferenciá-lo de outros sinais. As transitórias, também geradas pelo computador, são as que se modificam durante a vida do sinal. Elas simbolizam os diferentes estados que sua referência pode assumir.

Os sinais organizam a substância da expressão gerada pela execução de um programa. As transformações nesta substância refletem as ações dos sinais. Andersen propõe uma

arquitetura de sistemas estruturada como uma coleção de sinais computacionais, cuja especificação contém no mínimo duas partes; a semiologia e a meta-semiologia. A primeira descreve as características do sinal que são pertinentes para sua interpretação, manipulação e diferenciação. A meta-semiologia descreve como a semiologia é de fato implementada.

4.2.2 - Os sinais computacionais elementares

A tipologia de sinais computacionais proposta por Andersen possui 6 categorias, originárias de uma função combinatória dos tipos de propriedades de sua expressão. Assim são definidos os sinais interativos, os atores, os controles, os sinais objetos, os de layout e os fantasmas.

4.2.2.1 - Sinais interativos

O sinal interativo é o único tipo de sinal que apresenta o conjunto completo de propriedades permanentes, transitórias, manipuláveis e ainda executa ações.

Sinais interativos e os sinais objetos estabelecem uma relação do mesmo tipo da que se verifica entre uma ferramenta e um objeto de trabalho. Os sinais interativos denotam a ferramenta que o usuário manipula para influenciar um objeto de trabalho, aqui expresso pelo sinal objeto. O cursor de texto é o instrumento que o usuário manipula via teclado para modificar o campo de texto. Como acontece na vida cotidiana neste tipo de relação a concepção orientada sinal atribui a ação ao sinal interativo. No exemplo acima o cursor e não o campo de edição, como acontece no paradigma orientado objeto, é quem realizaria a ação de edição. Somente os cursores de texto, do mouse e os diferentes tipos de cursores para desenhos podem ser classificados como sinais verdadeiramente interativos.

4.2.2.2 - Sinais atores

Aos sinais atores faltam as características manipuláveis. Eles não podem ser influenciados diretamente pelo usuário mas podem influenciar outros sinais. Os sinais atores também participam de funções com sinais objetos, nas quais eles denotam máquinas ou ferramentas automáticas que trabalham sobre um objeto. O usuário não pode intervir no desenvolvimento do processo. Em um editor de textos por exemplo, as opções de repaginar, inserir índice e sumário, após iniciadas, não podem ser influenciadas pelo usuário. Ele deveria poder interrompê-las.

4.2.2.3 - Sinais controles

Os controles só possuem propriedades permanentes e suas ações são percebidas indiretamente ao influenciarem o comportamento de outros sinais. Os controles representam as restrições impostas a máquinas e ferramentas. Eles agem sobre as características transitórias de sinais interativos e atores. São tipicamente sub-áreas como as bordas de uma janela de texto que limita os caminhos do cursor de texto e modifica a forma do cursor do mouse. Quando o cursor está sobre uma sub-área ele tem a forma de um "I" e sobre outra tem a forma de uma flecha.

4.2.2.4 - Sinais objetos

Os sinais objetos possuem características permanentes e transitórias, são influenciados por outros sinais e não podem ser manipulados diretamente pelo usuário. Eles representam os objetos do trabalho que participam de funções com os sinais interativos, vistos como ferramentas e com os sinais atores, vistos como máquinas automáticas.

4.2.2.5 - Sinais figurativos e fantasmas

Os sinais figurativos só possuem propriedades permanentes, eles não têm função face aos outros sinais do sistema. Aos fantasmas faltam características dos três tipos. Eles são percebidos pelas ações que realizam sobre outros sinais.

4.2.3 - Os sinais computacionais compostos

Em um sistema interativo os sinais podem aparecer juntos ou se seguirem no tempo. O primeiro tipo de situação define uma cadeia concorrente que representa o ambiente estático de trabalho. Este é formado pelo elenco de objetos de trabalho, máquinas, ferramentas, controles, etc. A cadeia sequencial, definida pelo segundo tipo de relação representa o aspecto dinâmico do sistema. Elas representam as possibilidades e os padrões em termos de ações. Estes dois tipos de cadeias podem ser analisados em termos de sintagmas e paradigmas o que resulta em 4 combinações;

- sintagma concorrente - função entre elementos que ocorrem juntos em uma cadeia concorrente. Em um sistema informatizado estes sintagmas representam as cenas para as ações, com objetos, máquinas, ferramentas, etc...

- *paradigma concorrente* - representa o elenco de elementos que podem substituir um determinado elemento em sua mesma posição dentro de uma cadeia concorrente. Define objetos que possuem funções similares.
- *sintagma sequencial* - função entre elementos de uma cadeia sequencial. Eles representam a sintaxe das ações.
- *paradigma sequencial* - representa o elenco de elementos que podem substituir um determinado elemento em sua mesma posição dentro de uma cadeia sequencial. Define o paradigma de ações e tarefas.

Os sinais compostos são formados a partir de funções ou de relações entre sinais. Assim eles correspondem aos sintagmas concorrentes e sequenciais.

4.2.3.1 - Os concorrentes - cena, vista e foco

O principal sinal composto concorrente definido por Andersen refere-se à "cena". As cenas correspondem à noção teatral do termo, que define um local com os objetos e os atores necessários para a realização de ações. Ela exprime um sintagma concorrente entre sinais objetos e sinais interativos que é pressuposto pelo paradigma das tarefas. Leia-se o conjunto de objetos e ferramentas necessários para a execução de um grupo de tarefas concorrentes. Andersen sugere que a descrição de um sistema interativo baseada em cenas deve ser feita em dois níveis. O primeiro descreve cenas genéricas, como por exemplo aquelas ligadas ao gerenciamento do sistema de janelas, da manipulação de arquivos, dos dispositivos de entrada e saída, etc. Num segundo nível ocorre a descrição das cenas associadas às tarefas específicas de um aplicativo. Assim a concepção de um sistema interativo pode se realizar como um processo de inserção de novas cenas em um livro já escrito e comercializado, como aquelas definidas nos sistemas MS-Windows, X-Windows, MacApp, etc.

Outros objetos compostos concorrentes são a vista, a sub-vista e o foco. A sub-vista é uma relação ou um sintagma concorrente entre dois ou mais sinais que representam, em substâncias de expressão diferentes, o mesmo conteúdo. Ela exprime uma situação onde as modificações introduzidas pelo usuário sobre um sinal que representa uma determinada característica de um determinado objeto repercutem sobre outro sinal, que representa esta mesma característica deste mesmo objeto, mas de uma maneira diferente. As sub-vistas manifestam a capacidade particular que os sistemas informatizados têm de apresentar ao

usuário diferentes representações simultâneas e coerentes de um mesmo objeto. Isto ocorre em um aplicativo de desenho por exemplo, onde um objeto tem sua posição e/ou dimensão representada graficamente e textualmente. Ambas reagem de maneira conjunta às mudanças introduzidas pelo usuário. Todas as sub-vistas de um mesmo objeto definem um vista. O foco é uma relação entre dois ou mais sinais que representam, na mesma substância de expressão, diferentes partes do mesmo conteúdo. Ele se verifica em um editor de texto quando a janela do documento é repartida em duas. Cada sub-área representa textualmente diferentes partes ou a mesma parte do documento.

4.2.3.2 - Os sequenciais - tarefa, ação e indicador

Andersen rotula os sinais compostos sequenciais de "ações e tarefas simbólicas que resultam da manipulação de sinais". Estes sintagmas correspondem aos padrões na sequência dos atos simbólicos e são definidos a partir do procedimento analítico estruturalista. As maiores partes significativas resultantes da divisão de uma atividade são as tarefas e correspondem ao conceito linguístico de período "os menores elementos que podem funcionar como o todo do qual fazem parte". As ações são partes de uma tarefa e correspondem à noção de sentença "os maiores elementos que não podem funcionar como o todo do qual fazem parte". Enquanto as tarefas combinam livremente as ações dependem de uma tarefa. No exemplo utilizado por Andersen para diferenciar tarefa e ação, ele apresenta a cadeia de sinais verificada em um aplicativo de desenho; "seleciona ferramenta retângulo+desenha retângulo+seleciona caneta+desenha a mão livre". Esta sequência pode ser dividida em duas partes independentes "seleciona retângulo+desenha retângulo"+"seleciona caneta+desenha a mão livre". Elas formam tarefas pois podem funcionar como o todo do qual fazem parte. Na divisão da cadeia "seleciona retângulo+desenha retângulo" verifica-se que a segunda parte depende da primeira. Não podendo funcionar como o todo do qual faz parte, "seleciona retângulo" define uma ação. Um outro tipo de sinal sequencial proposto por Andersen é o "indicador". Ele é o último elemento que apresentando catálise, pode funcionar como tarefa. Desenhar o retângulo requer movimentos da ferramenta retângulo no papel. O sinal representando a ferramenta pode sozinho indicar ou "catalisar" que a tarefa de desenhar um retângulo está sendo realizada.

4.2.4 - O caos e o diálogo

Em publicações recentes este autor explora a visão de um sistema informatizado como meio elástico. Nele, ao contrário do que ocorre em livros ou em filmes, o usuário pode mudar o significado do sistema durante seu processo. Escrever nas margens de um livro ou arrancar suas folhas, faz parte da criação do seu significado, que pode ser considerado tanto como vandalismo como personalização. Um sistema informatizado permite a personalização, apesar de não ser completamente flexível. O projetista impõe uma estética que é representada por sua elasticidade, definida como a maneira com a qual a estrutura do sistema reage às ações do usuário. Para descrever o aspecto dinâmico de um sistema interativo Andersen (Andersen, 1992) propõe que seus sinais sejam encarados como vetores, sujeitos à forças dinâmicas de atração e de repulsão determinadas pela equação cúspide ($y = x^4 + ax^2 + bx$). Esta equação, uma das catástrofes elementares, tem a capacidade de representar mudanças quantitativas (grandeza dos mínimos) e qualitativas (um ou dois mínimos) que caracterizam as forças entre vetores. Assim, a teoria do caos é chamada a dar sua contribuição na importante questão do controle do diálogo de sistemas interativos.

4.3 - Conclusões

O esquema semiótico considerado neste trabalho envolve planos de conteúdos e de expressões em uma relação simbólica que se dá na mente de um interpretador. A partir do entendimento de um sistema interativo como um sistema de expressões cujo conteúdo está no sistema de referência, os sinais computacionais são definidos como sinais candidatos. Eles dependem do usuário para se realizarem como sinais. Entretanto o projetista, e é este o seu papel, deve poder influenciar sua interpretação.

A tipologia de sinais proposta neste capítulo baseia-se na combinação dos seus aspectos da expressão. Ela motiva comparações com algumas das relações verificadas no trabalho, envolvendo objetos, ferramentas e máquinas automáticas. Os sinais compostos são definidos a partir das relações entre sinais elementares concorrentes e sequenciais.

Este capítulo encerra a primeira parte da monografia dedicada à ergonomia e às interfaces homem-computador. Ela traz a revisão bibliográfica sobre os conhecimentos disponíveis e as formas de intervenção ergonômica na concepção de interfaces homem-computador. Estas interfaces são por sua vez, alvo de uma revisão sob o ponto de vista

computacional e sob o ponto de vista semiótico. Da interação entre estas diferentes perspectivas resulta a proposta inédita dos OIA^e, componentes lógicos de um sistema de geração de interfaces homem-computador que facilitam o raciocínio ergonômico.

Esta proposta é detalhada na segunda parte desta monografia, com a apresentação das classes de OIA^e e de seus atributos ergonômicos.

Segunda Parte - O modelo de OIA^e

A revisão bibliográfica realizada na primeira parte desta monografia inclui os elementos indispensáveis para o entendimento da proposta referente aos sinais integrantes de uma arquitetura lógica para a geração de interfaces ergonômicas.

Os OIA^e - Objetos de Interação Abstratos Ergonômicos - são definidos como facilitadores do raciocínio ergonômico sobre a configuração dos aspectos de utilização de um interface homem-computador. Existe uma correspondência efetiva entre os OIA^e e as recomendações ergonômicas que os revelam.

A noção de OIA^e é detalhada no capítulo V, que inclui a apresentação da morfologia e da tipologia destes elementos. Os capítulos seguintes VI, VII e VIII descrevem respectivamente as relações estruturais, formais e funcionais que definem as classes de OIA^e.

5 - DESENVOLVENDO O MODELO DE OIA^e

O objetivo neste capítulo é de introduzir a noção de OIA^e - Objeto de Interação Abstrato Ergonômico -, apresentando seus objetivos e descrevendo os passos trilhados no desenvolvimento do modelo que é descrito no seguimento desta monografia.

Inicialmente serão descritas suas finalidades, e de como elas definem a dupla abstração dos OIA^e. Uma definição geral é proposta e após aprofundada com a apresentação da morfologia de atributos e da tipologia de OIA^e. O capítulo se encerra com a apresentação dos detalhes referentes ao processo de identificação de exemplares destes objetos ou sinais, a partir das recomendações ergonômicas.

5.1 - Objetivos e definição

Os OIA^e se definem na arquitetura de uma sistema de geração de interfaces homem-computador ergonômicas. Eles representam o conjunto de componentes de interação de que dispõe o projetista para realizar uma interface e sobre os quais se referem boa parte das recomendações ergonômicas.

O primeiro objetivo de um OIA^e é a portabilidade. Ele visa proporcionar a independência das configurações geradas na etapa de projeto face aos diversos ambientes de concepção considerados. A abstração que se impõe resulta da proliferação de toolkits acompanhada de uma inflação de objetos de interação concretos. Objetos abstratos definem recursos abstratos que podem ser realizados através de correspondências com os recursos concretos de um ambiente de concepção específico.

O segundo objetivo refere-se ao raciocínio ergonômico. Um OIA^e é definido de maneira a maximizar as correspondências entre seus atributos e serviços com as regras ergonômicas. A abstração neste caso, se refere a suas origens nas próprias recomendações ergonômicas. O modelo de OIA^e desconsidera assim, os recursos ligados à lógica de funcionamento dos sistemas, vistos como desnecessários ao raciocínio ergonômico. Ao mesmo tempo eles propõem novos recursos, sem similares nos objetos de interação concretos e que

estão ligados à lógica de utilização de uma interface. Estes novos recursos enriquecem a noção de objeto de interação no sentido da criação e da manutenção das qualidades ergonômicas de uma interface homem-computador.

Um OIA^e é abstrato pois pode ser implementado através dos diversos tipos de objetos de interação concretos existentes. Ele é abstrato ergonômico pois é identificado nas recomendações ergonômicas para facilitar o raciocínio por regras ergonômicas.

Os OIA^e são objetos interativos independentes face aos diversos ambientes de concepção que, definidos a partir do exame de recomendações ergonômicas têm o objetivo de facilitar o raciocínio ergonômico sobre a concepção/avaliação das interfaces homem-computador".

5.2 - Materiais e métodos

O principal material utilizado no desenvolvimento do modelo de OIA^e são recomendações ergonômicas. Embora com objetivos diferentes, elas são usadas nas duas etapas de atividades que marcam este trabalho: as definições estruturais e a identificação de exemplares de OIA^e.

5.2.1 - Etapa de definições estruturais

Antes de que se possa identificar os exemplares de OIA^e nas recomendações ergonômicas é necessário que se aprofunde o entendimento sobre a natureza de componentes e de um modelo de objetos voltados para a lógica de utilização de sistemas interativos. Esta é composta de significados e de relações entre sinais.

As noções teóricas da semiótica fornecem assim, os requisitos para a busca de uma técnica de engenharia de software interativo capaz de apoiar a noção de sinal computacional derivado das recomendações ergonômicas. Toda a proposta gerada em termos de morfologia de OIA^e é testada através da verificação do nível de correspondências com um conjunto de 200 recomendações ergonômicas definidas aleatoriamente.

A cada recomendação ergonômica examinada as hipóteses em termos de morfologia e de tipologia de OIA^e ficam sujeitas a modificações que afetam as interpretações seguintes e anteriores. Em seus comentários sobre a hermenêutica, Winograd e Flores (Winograd &

Flores, 1986) colocam que significado de um texto, não necessariamente sagrado, é algo que depende da interação entre o horizonte fornecido pelo próprio texto e o horizonte que o interpretador traz para ele. O "ciclo hermenêutico" é inevitável pois o que compreendemos é uma função do que já sabemos, e o que sabemos é algo que se modifica a cada novo entendimento. Assim, o ciclo hermenêutico se instala até que o modelo de OIA^e se estabilize frente as recomendações ergonômicas.

As propostas descritas a seguir representam o resultado do exame de cerca de 200 recomendações registradas na "Base de recommandations ergonomiques pour l'évaluation et la conception d'interfaces utilisateurs" (Aschehoug & Scapin, 1990) que é mantida pelo Projeto de Psicologia Ergonômica para a Informática do Inria. Estas primeiras serviram de controle para a etapa de definição da morfologia de um OIA^e e das hierarquias de classes.

5.2.1.1 - A morfologia de um OIA^e

As ferramentas informatizadas apresentam uma natureza essencialmente simbólica. Seus componentes são sinais que o usuário detecta, interpreta e manipula para agir sobre outros sinais. A característica básica de um sinal é a correspondência, que ocorre na mente do interpretador, entre um plano de conteúdo e um plano de expressão.

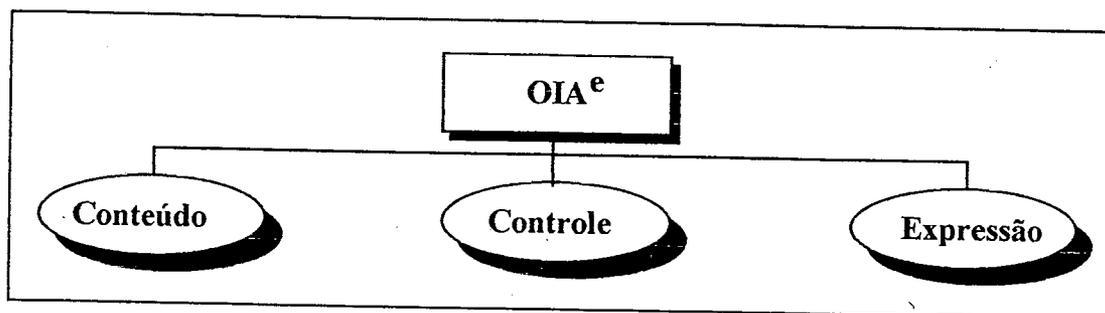


Fig 5.1 - A morfologia de um objeto de interação abstrato ergonômico.

Um OIA^e é construído na analogia com um sinal informatizado e apoia-se na noção de agente PAC. Ele realiza, através da dimensão de controle, a correspondência entre um conteúdo que se deseja transmitir e uma expressão definida para este fim. Trata-se de um "sinal candidato", como aliás o são todos os sinais naturais ou artificiais. O que o distingue de outros tipos de sinais é o fato de que seu conteúdo, mesmo que escondido do usuário, reside explícito dentro do meio computacional.

Sob o ponto de vista da correspondência com as recomendações ergonômicas, verifica-se a vantagem da adoção de tal morfologia, pois diversas delas deliberam sobre os conteúdos a veicular e sobre a maneira de fazê-lo.

A expressão é a dimensão visível do sinal, ela possui os recursos para apresentar informações e receber as ações do usuário. O conteúdo é a mensagem se quer transmitir ao usuário. O controle é o gerente que garante a coerência entre conteúdo e expressão.

As funções do controle incluem a codificação e estruturação. A primeira se encarrega de manter a coerência entre o estado do conteúdo e estado da expressão e a segunda de gerenciar as relações com outros objetos dos sistema. Em tempo de execução o controle define o tipo de expressão que o sinal deve apresentar em função de seu conteúdo candidato corrente. O controle deve fazê-lo também para as modificações realizadas pelo usuário sobre a dimensão apresentação. As composições são definidas a partir de relações entre objetos e neste caso, o controle da composição garante o gerenciamento destas relações.

Os OIA^e modelizam as classes de componentes de uma ferramenta genérica aos quais os objetos do núcleo funcional se associam para se comunicarem com o usuário. Assim, muitos de seus conteúdos candidatos podem ser identificados a partir de sua função na ferramenta e de seu papel na sintaxe do sistema.

5.2.1.2 - A tipologia de OIA^e

Os OIA^e são sinais computacionais, que apresentam uma estrutura tridimensional de atributos de conteúdo, de controle e de expressão. A estes tipos de atributos podem ser associadas as relações funcionais, estruturais e formais que se estabelecem entre eles. Assim, todo e qualquer OIA^e desempenha um papel na interação, integra ou possui uma estrutura e apresenta uma forma particular e diferenciável.

A tipologia proposta apresenta três perspectivas do mesmo objeto, cada uma enfocando um tipo de relação específica.

As classes de estruturas são definidas a partir das funções estruturais desempenhadas pelo controle de um OIA^e. Uma estrutura pode ser elementar ou composta. Da divisão de uma composição resultam componentes. Da divisão de um elemento resultam partes. Ao contrario

dos componentes, as partes nunca ocorrem isoladamente. A composição representa um tipo de container que assegura a coesão espacial de seus componentes. Um tipo especial de composição gerencia as restrições em termos da expressão dos sinais que participam da composição. A correlação assegura relações envolvendo características do conteúdo de um objeto do núcleo funcional e características da expressão de um objeto de interação.

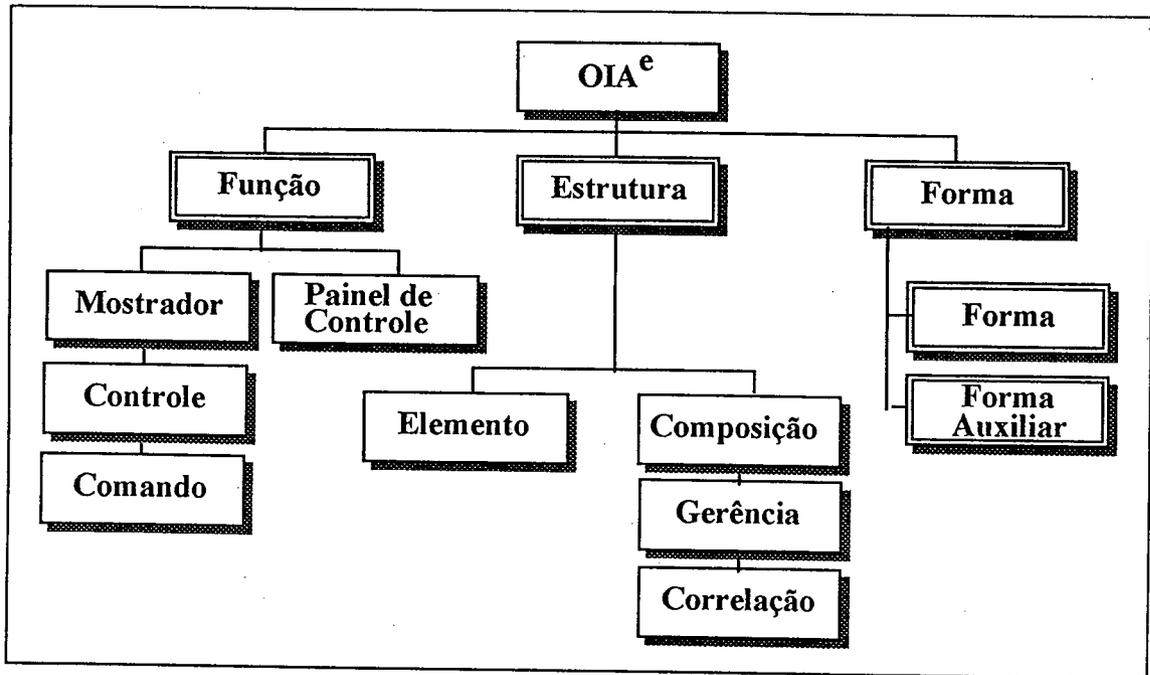


Fig 5.2 - As Classes de Objetos de interação abstratos ergonômicos

As classes de formas baseiam-se nos aspectos da expressão de um OIA^e. Elas são divididas segundo formas básicas e formas auxiliares. O conjunto de formas básicas correspondem às escolhas em termos de formas de expressão possíveis para uma estrutura. Estas escolhas são independentes umas das outras. As formas auxiliares visam completar a carga informacional da forma básica à qual está ligada. As formas auxiliares nunca ocorrem isoladamente.

A classificação funcional se baseia nas relações de um OIA^e com o usuário e com a aplicação. As classes aqui definidas modelam os mostradores, controles e comandos que compõem uma ferramenta genérica. Os mostradores são acionados pela aplicação para apresentar dados e informações ao usuário. Os controles são mostradores que podem ser acionados pelos usuários para a entrada dos parâmetros de uma ação. Os comandos são controles que têm a capacidade de disparar as ações do núcleo funcional. Os mostradores são

considerados como a classe mãe controles e comandos. Os painéis de controle proporcionam um elenco de funções de mostradores, controles e comandos necessários à realização de ações e tarefas.

A hierarquia funcional se apoia na disseminação das metáforas como princípio básico para a concepção de interfaces homem-computador. Através de metáforas os projetistas exploram as lógicas de funcionamento e de utilização desenvolvidas pelos usuários de dispositivos não informatizados e largamente difundidos nos ambientes profissionais. Assim eles buscam conseguir a usabilidade de seus sistemas. O exemplo clássico é o da metáfora "desktop" ou da mesa de escritório, que propõe os objetos de sistemas informatizados à semelhança dos componentes de um escritório. Este princípio básico é verificado em aplicações de edição de textos ou de som que apresentam os mesmos mostradores, controles e comandos de uma máquina de escrever, de um gravador ou de uma mesa de som. Mesmo que sejam mais abrangentes do que suas primas manuais, mecânicas ou eletrônicas, as ferramentas informatizadas fundamentam-se nas metáforas de seus mecanismos básicos. A utilização de metáforas de ferramentas se dissemina como princípio de concepção, assim é natural que os componentes disponíveis em uma "toolbox" estejam voltados para a concepção de ferramentas genéricas. As classes de funções que compõem os dispositivos não informatizados em geral e que podem vir a ser utilizados para as metáforas computacionais são mostradores, controles, comandos e painéis de controle.

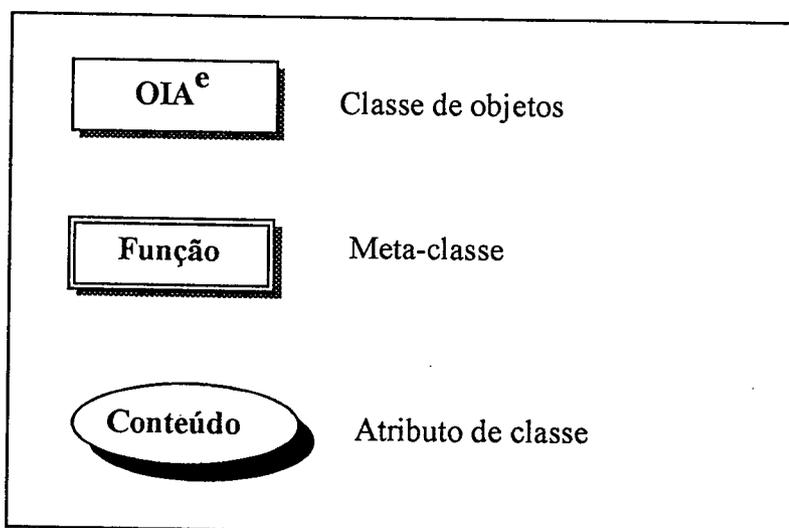


Fig 5.3 - Simbologia utilizada na apresentação do modelo de OIA^e

Os elementos da apresentação do modelo de OIA^e podem ser classificados como classes de objetos, meta-classes e atributos de classe. As meta-classes têm a finalidade de organização do modelo. Conforme foi apresentado no capítulo III, elas não definem entidade alguma do domínio, sendo comuns a diversas outras classes que efetivamente se realizam. O simbolismo ligado a estes elementos está descrito na figura 5.3.

5.3 - A identificação dos OIA^e

Os exemplares das classes de OIA^e e das classes dos atributos que povoam as três perspectivas de sinais são identificados nas recomendações ergonômicas. Face ao nível de abstração pretendido eles não incluem todas as classes e os atributos possíveis, mas somente aqueles ligados à utilização dos sistemas. Diferentemente de outras modelagens ergonômicas voltadas para a especificação ou para a descrição de interfaces homem-computador, o modelo de OIA^e está voltado para a configuração destas interfaces. Enquanto a especificação pode envolver qualidades como consistência e homogeneidade, a configuração envolve características como dimensões, cores, etc. Estes são os aspectos diretamente ligados ao raciocínio por regras sobre a utilização dos sistemas interativos.

Os objetos e atributos identificados nesta etapa de atividades definem os recursos de que dispõe o projetista para configurar a utilizabilidade de seus sistemas. As 500 recomendações aqui examinadas têm origem distintas (Bodart & Vanderdonckt, 1993b; Brown, 1988; Rivlin, et al., 1990; Shneiderman, 1987; Smith & Mosier, 1986; Valentin & Lucong Sang, 1987; Williges & Williges, 1984).

5.4 - Resumo do capítulo

✓ Os OIA^e são objetos interativos independentes face aos diversos ambientes de concepção que, definidos a partir do exame de recomendações ergonômicas têm o objetivo de facilitar o raciocínio ergonômico sobre a concepção/avaliação das interfaces homem-computador.

✓ A facilitação é conseguida através das correspondências entre atributos e serviços de OIA^e e as recomendações ergonômicas que compõem a base de conhecimento de um sistema de geração de interfaces homem-computador.

✓ Um OIA^e é construído na analogia com um sinal informatizado e apoia-se na noção de agente PAC. Ele realiza, através da dimensão de controle, a correspondência entre um conteúdo que se deseja transmitir e uma expressão definida para este fim.

✓ Todo e qualquer OIA^e desempenha um papel na interação, integra ou possui uma estrutura e apresenta uma forma particular e diferenciável. A tipologia proposta baseia-se nestas relações e define as perspectivas funcional, estrutural e formal do modelo de OIA^e

Os recursos que o modelo de OIA^e propõe para a configuração da usabilidade das interfaces homem-computador são descritos nos capítulos seguintes.

6 - A PERSPECTIVA ESTRUTURAL

Sob o ponto de vista estrutural, as relações entre os OIA^e definem duas classes de estruturas: os elementos e as composições. Da divisão de uma composição resultam componentes, que podem funcionar como o todo que integram. Um componente pode ser um elemento ou uma composição. As divisões de um elemento produzem partes. As partes não podem funcionar como o todo que integram. Em outras palavras, as partes resultantes da divisão de um elemento nunca ocorrem isoladamente.

A composição assegura a coesão espacial de seus componentes. Um tipo especial de composição, a gerência, impõe características específicas da expressão de seus componentes. A correlação é uma classe de gerentes que assegura as relações entre participantes genéricos de uma composição, não necessariamente objetos de interação.

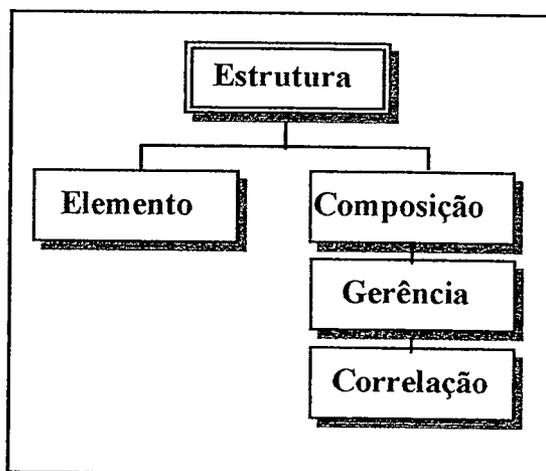


Fig 6.1 - As classes de Estruturas

Tanto uma composição como um elemento possuem atributos estruturais que herdam da meta-classe Estrutura.

6.1 - A classe Estrutura

O conteúdo da classe "Estrutura" é identificado nas recomendações ergonômicas e refere-se ao atributo genérico "demanda_de_atenção" que o objeto requer em um determinado momento na tela. Os valores possíveis deste atributo, também identificados nas recomendações são; demanda de atenção "crítica", "urgente", "imediate", "particular" ou

"casada". A demanda de atenção "particular" se aplica quando um objeto deva ser discriminado na tela e a "casada", quando ele deva ser associado a outros objetos.

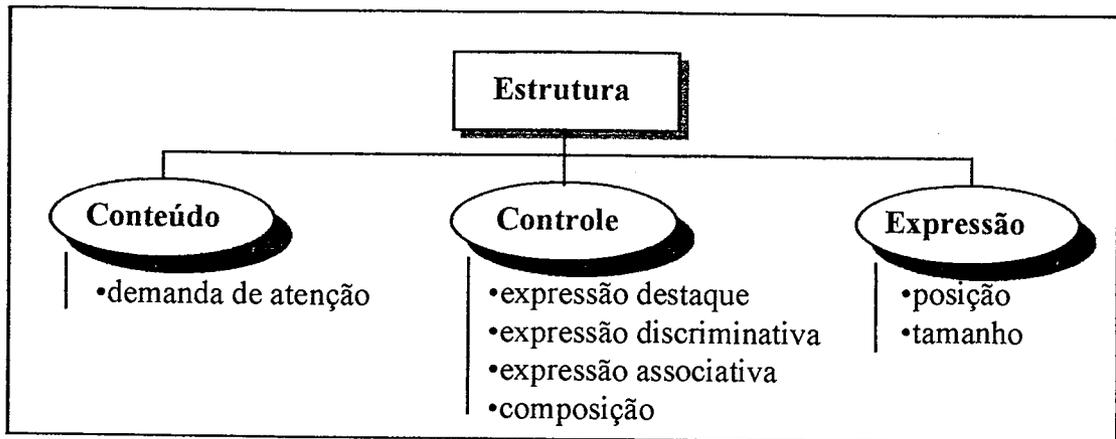


Fig 6.2 - A classe Estrutura

O controle de uma estrutura deve garantir a coerência entre seu conteúdo e sua expressão. Assim seus atributos são definidos de forma que o projetista possa, em tempo de projeto, configurar as ligações entre um tipo de conteúdo e um tipo de expressão. Ele o faz se valendo dos atributos "expressão_destaque", "expressão_discriminativa" e "expressão_associativa".

A "expressão_destaque" associa uma demanda_de_atenção "crítica", "urgente" ou "imediate" a uma lista de atributos de expressão a ser definida pelo projetista. Em tempo de execução cabe ao controle do objeto acionar as expressões predefinidas toda a vez que o objeto se tornar crítico, urgente ou demandar atenção imediata.

O mesmo ocorre quando um OIA^e participa de uma composição com outros objetos e, por determinação do gerente da composição, ele esteja associado a outros objetos ou deva ser diferenciado de todos. As expressões de controle podem ser formadas por qualquer combinação de atributos de expressão incluindo as formas básicas e as auxiliares como cor, textura, estilo, posição, tamanho, orientação, etc.

Qualquer estrutura, elementar ou composta, pode participar de uma composição. Este fato reflete-se no atributo de controle que guarda o nome da "composição" de que ela é componente.

6.2- A classe Elemento

Um elemento herda as características de uma estrutura e propõe os recursos relativos à noção de partes. As partes de um objeto elementar variam de ambiente para ambiente, mas via de regra são definidos um primeiro plano, um plano de fundo e bordas. Enquanto o primeiro plano recebe as palavras e ícones, o plano de fundo recebe os motivos e sombras.

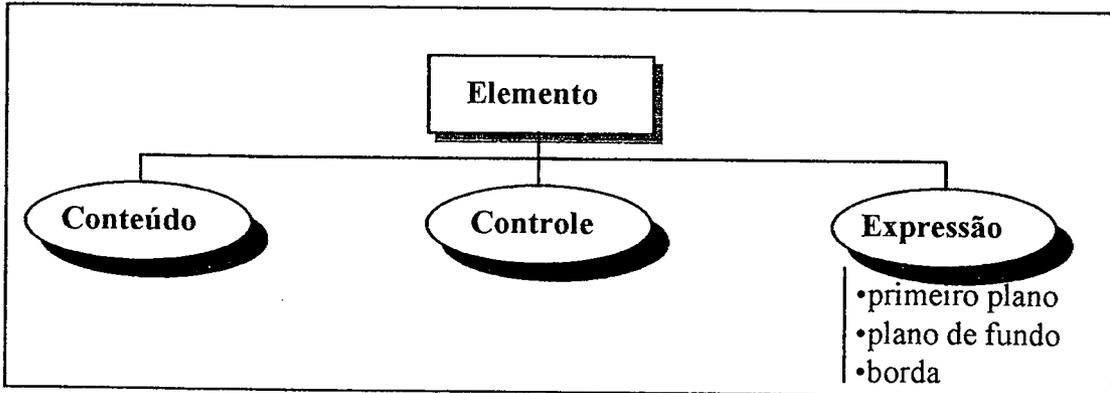


Fig 6.3 - A classe Elemento

6.3 - A classe Composição

Uma Composição apresenta os atributos de controle "lista de componentes" e "layout". Os objetos desta classe desempenham o papel de containers ao garantir a coesão espacial na apresentação, deslocamentos e eliminação da tela dos objetos declarados na "lista de componentes". Estes itens têm os valores default para seus atributos definidos a nível da composição. Assim, se determinado estilo de caracteres for definido na composição, todos os seus participantes, na ausência de uma declaração explícita, terão esta característica de estilo configurada.

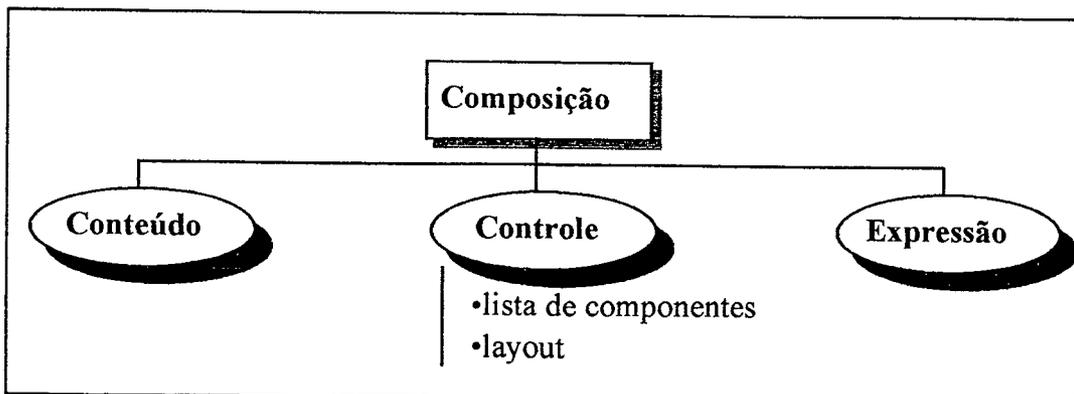


Fig 6.4 - A classe Composição

As possibilidades em termos de layout dos componentes de uma composição incluem o layout livre, o alinhamento, o balanço, a simetria e a grade. O layout livre posiciona os elementos de uma composição a partir de tabelas de coordenadas. As opções de alinhamento, balanço e simetria se baseiam em uma direção definida para alinhar, distribuir equidistantemente ou opor componentes. A grade posiciona elementos segundo um número de linhas e colunas definidas.

6.4.1 - A classe Gerência

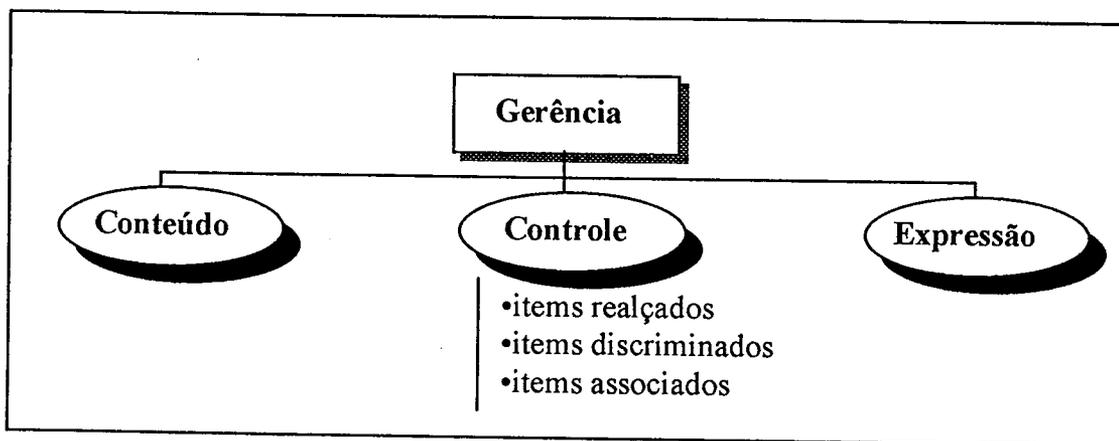


Fig 6.5 - A classe Gerente de Composição

Os objetos desta classe propõem recursos para gerenciar as expressões de controle ao nível dos componentes declarados nas listas "itens_associados"; "itens_discriminados" e "itens_realçados". O gerente envia uma instrução modificando o valor do atributo de conteúdo "demanda de atenção" nos objetos declarados nestas listas, que se encarregam eles mesmos, de atualizar suas expressões. Através deste dispositivo a comunicação entre objetos torna-se concisa e significativa.

Os projetistas podem explorar a possibilidade relativa às declarações de valores "par défaut" apresentadas no parágrafo anterior para definir expressões genéricas para todos seus componentes. Isto favorece a consistência das interfaces ao mesmo tempo em que facilita a tarefa de programação.

No caso de associações, o projetista pode configurar uma expressão associativa através da cor de fundo amarelo, mesmo antes de saber quais os elementos que devem ser associados. Ao serem criados, os elementos que participarem desta composição herdam, na ausência de uma declaração explícita, os valores definidos para expressão associativa no âmbito da composição.

O mesmo ocorre para as expressões discriminativas, que não podem ser as mesmas definidas para associações.

6.4.2 - A classe Correlação

Entre os recursos estruturais associados ao modelo de OIA^e consta a possibilidade de correlacionar características de objetos do núcleo funcional com características de objetos de interação. A classe "correlação" realiza este tipo de associação através de equações correlacionado as características de seus componentes. A existência deste recurso no modelo de OIA^e garante o funcionamento de classes de mostradores de dados como o "termômetro". A correlação associa o valor da variável do núcleo funcional "temperatura" com a altura de uma barra vertical representando uma coluna de mercúrio e/ou com um valor numérico. Esta classe representa um recurso de funcionamento que é previsto no modelo de OIA^e como forma de realizar recursos de utilização.

6.6 - O resumo do capítulo

- ✓ Sob o ponto de vista estrutural, as relações entre os OIA^e definem duas classes de estruturas: os elementos e as composições.
- ✓ Da divisão de uma composição resultam componentes. Da divisão de um elemento resultam partes. As partes não ocorrem isoladamente.
- ✓ O conteúdo da classe-mãe Estrutura apresenta o atributo genérico "demanda_de_atenção". Seu controle se vale dos atributos "expressão_destaque", "expressão_discriminativa" e "expressão_associativa". As expressões de controle podem ser formadas por qualquer combinação de atributos de expressão incluindo as formas básicas e as auxiliares como cor, textura, estilo, posição, tamanho, orientação, etc.
- ✓ Um Elemento apresenta as partes genéricas, primeiro plano, plano de fundo e bordas.
- ✓ Uma Composição apresenta os atributos de controle "lista de componentes" e "layout".
- ✓ O papel de gestor de relações entre objetos confere à "Gerência" os atributos de controle; "items_associados"; "items_discriminados" e "items_realçados". Estes atributos

acionam os recursos do gerente para instanciar o atributo "demanda de atenção" entre os objetos declarados nestas listas.

✓ A classe "correlação" realiza a associação através de equações correlacionado as características de um objeto da aplicação com as de um OIAe.

7 - A PERSPECTIVA FORMAL

A ocorrência de um sinal envolve a associação entre formas de conteúdo e formas de expressão. As formas de expressão resultam da articulação de substâncias de um contínuo perceptível ao sistema cognitivo humano. No modelo de OIA^e as formas de expressão incluem as formas básicas e as formas auxiliares. Estas classes estão definidas a partir da relação de subordinação que as envolvem. As formas básicas correspondem ao paradigma de formas independentes disponíveis para a expressão de uma estrutura. As formas auxiliares completam as possibilidades em termos de transmissão de informação de uma forma básica. Assim, a escolha de uma forma auxiliar é determinada pela forma básica predefinida.

7.1 - As classes de formas básicas

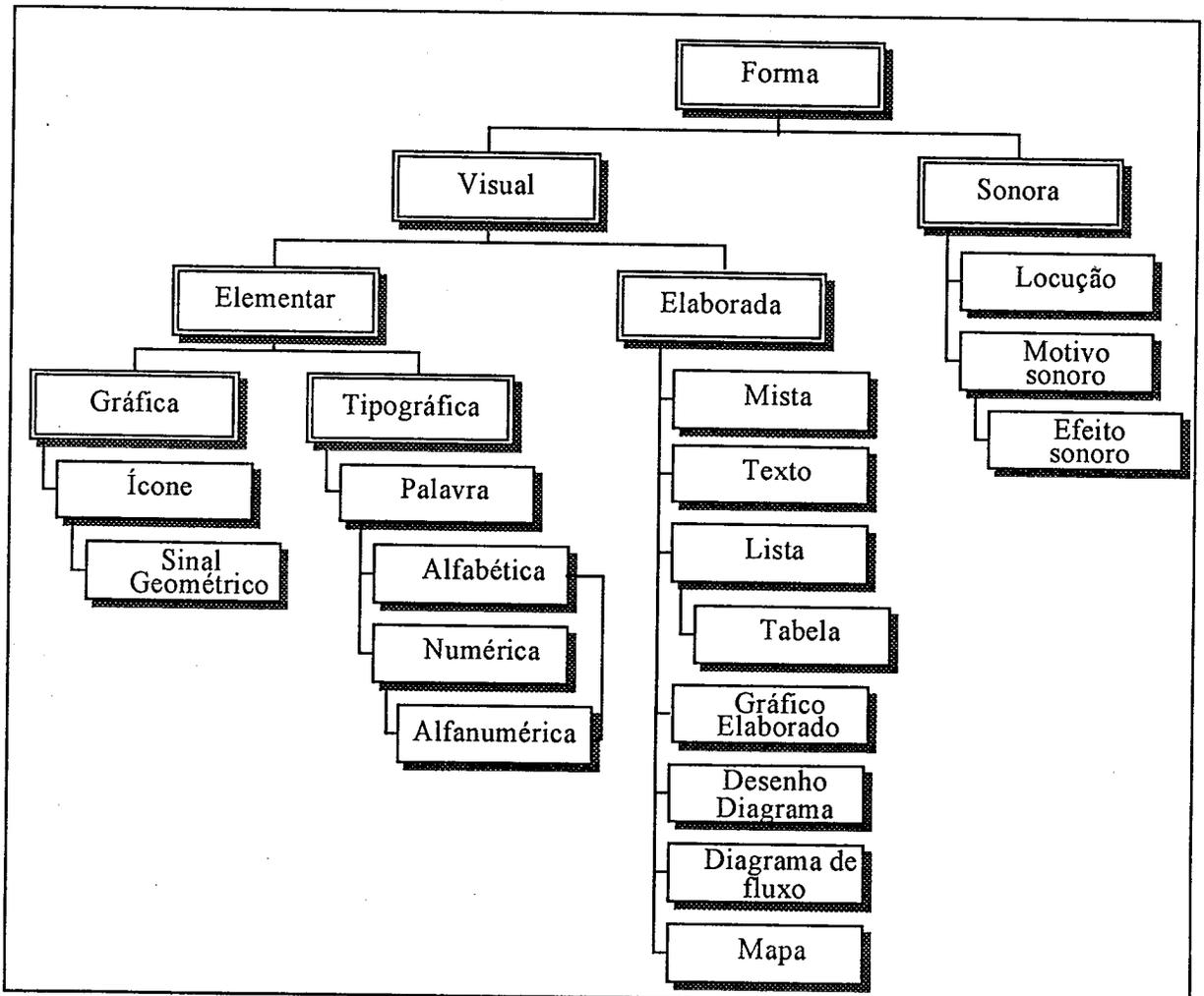


Fig 7.1 - As classes de Formas

A primeira divisão entre as formas básicas opõe sinais visuais e sinais sonoros. Os sinais sonoros articulam as substâncias de locução, motivos melódicos e o efeitos sonoros. Os sinais visuais podem ser elementares ou elaborados. Os elementares incluem os gráficos e os tipográficos. As representações gráficas incluem ícones e formas geométricas. As tipográficas incluem, as palavras alfabéticas, numéricas e alfanuméricas. As formas elaboradas, correspondem a um arranjo de formas elementares e incluem as formas mistas, os textos, as listas, as tabelas, os gráficos, os diagramas, as malhas e os mapas

7.1.1 - As formas sonoras

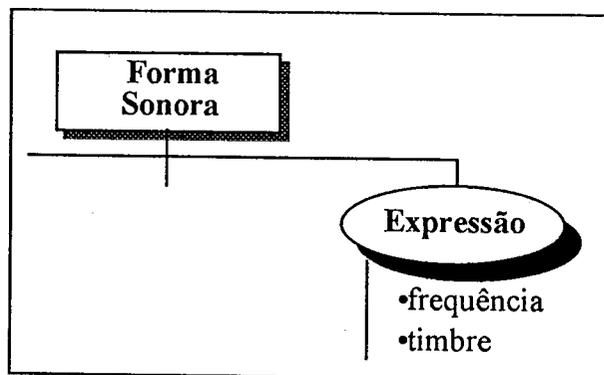


Fig 7.2 - A classe "Forma Sonora"

A meta-classe "Forma Sonora" apresenta os atributos de expressão "timbre", e "frequência", utilizados para destaque ou diferenciação do sinal sonoro. O timbre está ligado à natureza da entidade física que gera um som. A mesma nota musical em um piano ou em clarinete soam diferente devido a seus timbres particulares. A frequência também denominada registro de um som, pode ser alta ou baixa relativamente às oitavas. Aconselha-se a utilização de tons da mesma oitava para evitar problemas de construção de sinais sonoros

7.1.1.1 - A Locução

A classe "locução" representa uma forma alternativa de transmissão de mensagens ao usuário. Ela pode ser usada para demonstrações, simulações ou instruções e devem envolver somente as informações, críticas ou necessárias, que tenham um caráter de domínio público. Elas se aplicam aos ambientes de trabalho silenciosos, onde o operador tenha necessidades de pequenos deslocamentos. Uma locução pode ser de "fonte" sintetizada ou registrada. Se o conjunto de frases for em pequeno número elas devem ser registradas. Se elas foram sintetizadas deve-se tomar cuidado em definir um ritmo adequado. O timbre de voz pode ser

usado para destacar ou discriminar a expressão falada que não deve ser muito longa. Um meio adequado de permitir a interrupção da locução deve estar disponível ao usuário.

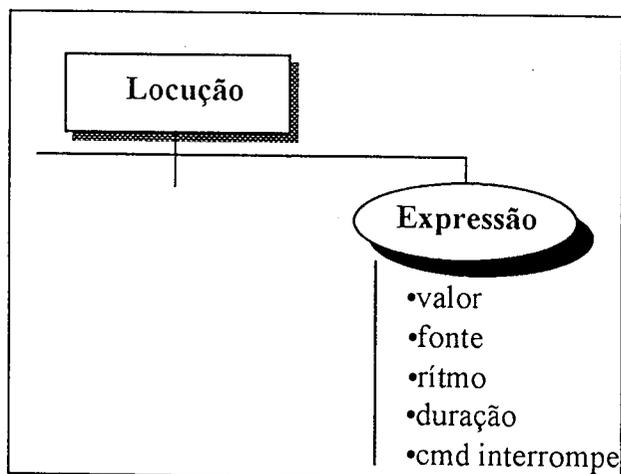


Fig 7.3 - A classe "Locução"

7.1.1.2 - O Motivo Melódico

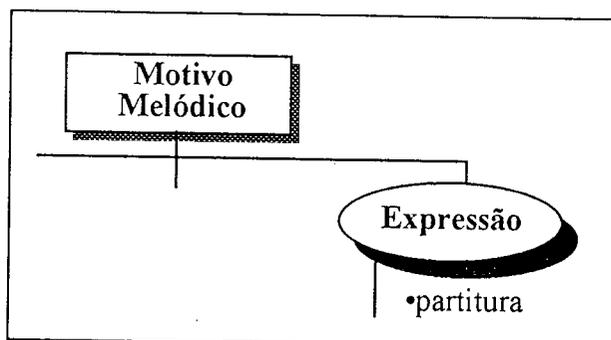


Fig 7.4 - A classe "Motivo Melódico"

Um "Motivo melódico" é uma breve sucessão de tons combinados de maneira a produzir um padrão de ritmo suficientemente distinto para que ele funcione como uma entidade individual e reconhecível (Loshe, Walker, Biolsi, & Rueter, 1991). A sequência de tom e o ritmo definem o atributo "partitura" de um "motivo sonoro". As características dinâmicas de "crescendo" e "decrescendo" podem ser utilizadas para indicar ações de iconificação, movimentação e mudanças de tamanho de janelas, por exemplo. Ritmos e de timbres são bastante eficazes para efeitos de diferenciação.

7.1.1.3 - O Efeito Sonoro

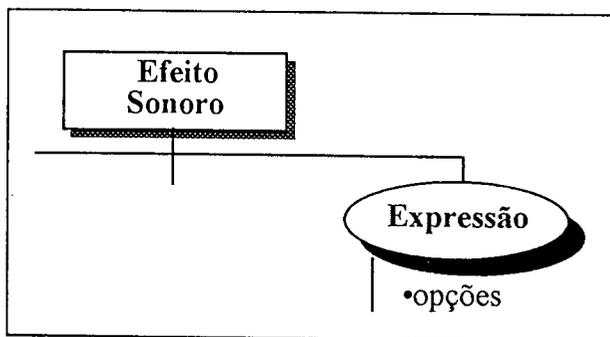


Fig 7.5 - A classe "Efeito Sonoro"

Os efeitos sonoros são estruturas simples que atuam como ícones sonoros para chamar a atenção do usuário e fornecer feedback. Eles devem ser consistentes com a situação a assinalar, pouco numerosos e suficientemente diferenciáveis. Como ícones sonoros, os efeitos sonoros podem apresentar formas concretas e abstratas. As concretas imitam sons como o fechamento de uma porta metálica para representar o fechamento de um arquivo de dados. As abstratas se baseiam em convenções sociais como aplausos para representar uma aprovação.

7.1.2 - As formas visuais

As formas "visuais" articulam substâncias sensíveis ao sistema visual humano e incluem as formas elementares, gráficas e tipográficas, e as elaboradas.

7.1.2.1 - As formas gráficas elementares: o ícone

Um ícone corresponde a um símbolo gráfico (Prieto., 1972). A primeira vista existe uma correspondência com o que se entende por ícone em computação. Uma análise mais aprofundada desta questão revela porém, um problema de interferência conceitual relacionando ícones, símbolos e sinais. Esta é a constatação possível após o exame de diversos trabalhos propondo conceituações e terminologias diferentes.

Considera-se que em um sinal (Chevalier, 1980) a relação entre a forma de conteúdo e forma de expressão seja arbitrária. Os sinais usados na álgebra e na matemática tem assim uma capacidade de transmitir um conhecimento objetivo. Em um símbolo, a homogeneidade entre expressão e conteúdo estabelece uma representação motivada ou concreta, onde o caráter espontâneo da interpretação é essencial. Estes autores salientam a capacidade que o símbolo

tem de deflagrar um efeito de ressonância. Ele representa conhecimentos cuja assimilação provoca um aprofundamento da existência dos indivíduos. Rasmussen confirma o efeito de ressonância que caracteriza um símbolo ao associá-lo com os conhecimentos necessários para raciocinar e planejar a condução da tarefa em situações complexas (Rasmussen, 1983). A citação de Chevalier (pag XV) é oportuna para concluir este entendimento: "o símbolo é carregado de realidades concretas. A abstração esvazia o símbolo e dá vida ao sinal".

De acordo com as considerações de Prieto, um ícone corresponde a um símbolo, portanto concreto, cuja expressão é uma imagem gráfica (Prieto., 1972). Nos estudos sobre sua utilização em interfaces homem-computador, os ícones são entretanto, considerados tanto como sinal abstrato, como símbolo concreto (Blattner, Sumikawa, & Greenberg, 1989; Green & Barnard, 1990). Rogers (Rogers, 1989) explora esta abrangência conceitual para propor quatro níveis de ícones; ícones de semelhança, ícones de exemplar, ícones simbólicos e ícones arbitrários. Tratam-se de considerações oportunas baseadas entretanto, em uma terminologia confusa. Chevalier, em seu Dicionário dos Símbolos esclarece sobre variações simbólicas pertinentes ao projeto representações gráficas:

Símbolo

representações gráficas motivadas ou concretas; um desenho de uma impressora para designar o dispositivo físico. Neste grupo se incluem as miniaturas.

Emblema

uma figura adotada convencionalmente para representar uma idéia, um ser físico ou moral; bandeiras nacionais e logomarcas

Atributo

um acessório característico para designar o todo; garfo e da faca para representar um restaurante, asas para companhias aéreas, rodas para o transporte rodoviário, etc

Arquétipo

exemplares de classe para representar o conjunto ; na régua do aplicativo de texto o nome "helvética", designa o conjunto de fontes dos quais o usuário dispõe..

Analogia

relação entre seres ou noções essencialmente diferentes , mas semelhantes sob um determinado aspecto ; taça de vinho rachada usada como símbolo de fragilidade.

Sinal

representações gráficas convencionadas ou arbitrárias.

No modelo de OIA^e um ícone corresponde a uma forma visual gráfica de caráter elementar. Seu valor gráfico pode variar, de acordo com as necessidades do projetista, entre o abstrato e o concreto. Apesar da interferência conceitual que ela representa, esta definição se justifica pois nível de abstração, uma qualidade abstrata, não faz parte deste modelo e nem pode ser fator de diferenciação entre classes de objetos. Em vista disto, a decisão cabível refere-se a que nome dar a esta classe de representações visuais. Por uma questão de coerência com os objetivos de uma proposta voltada para analistas, programadores e sua terminologia usual a classe dos OIA^e identificados como expressões visuais gráficas e elementares recebe o nome de ícone.

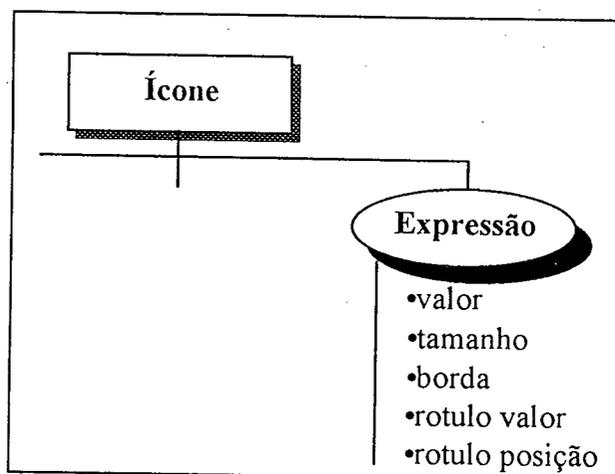


Fig 7.6 - A classe "Ícone"

Os Ícones devem ser significativos, apropriados, coerentes, consistentes, claros, simples e definidos em pequeno número (não mais do que 20). Seu tamanho deve ser econômico em relação ao espaço de tela. Alguns autores aconselham bordas bem definidas.

Diversos autores indicam a necessidade de um rótulo identificativo centrado na margem inferior. A presença deste rótulo faz com que, numa interpretação purista, a forma ícone mude de posição na tipologia e passe a corresponder à uma forma elaborada mista.

Os sinais geométricos correspondem a uma da classe de ícones abstratos que não apresentam labels ou bordas. Seus valores são predefinidos e aconselha-se não utilizar mais do que 15 opções de sinais geométricos.

7.1.2.2 - As formas tipográficas elementares: as palavras

As "Palavras" são representações visuais tipográficas que se valem dos caracteres editáveis. As expressões alfabéticas utilizam somente as letras, as numéricas, somente os números, os sinais das quatro operações (- + * /) e das pontuações numéricas (vírgula e ponto). As expressões alfanuméricas valem-se de letras, números, sinais de operações, de pontuação em geral (dois pontos, ponto e vírgula). As considerações sobre os valores a adotar para estas representações indicam não utilizar valores maiores do que 4 ou 5 caracteres para expressões arbitrárias em geral. Se isto for inevitável deve-se recorrer à pontuações e sub-grupamentos. O limite para expressões alfabéticas arbitrárias é de 7 a 10 letras. No caso de expressões alfanuméricas deve-se procurar arranjar letras e números em grupos separados, como forma de evitar combinações perigosas entre caracteres semelhantes.

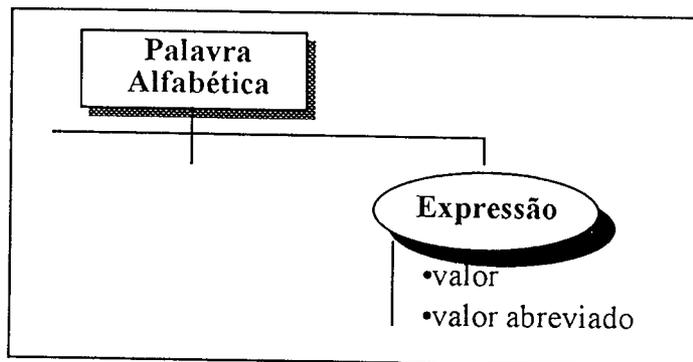


Fig 7.7 - A classe "Palavra Alfabética"

As palavras alfabéticas permitem a definição de uma, e somente uma, abreviatura, a ser utilizada quando absolutamente necessário. Ela deve ser curta, clara, significativa e não admite pontuações. As classes de objetos com esta forma podem assim reagir afirmativamente a uma solicitação de redimensionamento proposta por algum gerente de composição.

7.1.2.3 - As formas gráficas elaboradas:

Através das formas gráficas elementares os projetistas podem elaborar, de acordo com suas necessidades, diversas outras classes de recursos gráficos. As mais importantes, sob o ponto de vista ergonômico são o texto, a lista, a tabela, o gráfico, o diagrama, a malha e o mapa.

O "ícone com um rótulo alfanumérico" é considerado como uma forma elaborada mista, resultante da combinação de uma forma elementar gráfica com uma outra tipográfica.

7.1.2.3.1 - O Texto

A classe "Texto" apresenta informações na forma de diversas linhas de caracteres. O atributo "valor" guarda a cadeia de caracteres, que pode corresponder a um conjunto de frases e períodos. O formato de um texto pode ser normal ou em colunas e o recurso de hifenização, que autoriza a separação automática de palavras, é algo a evitar.

A justificação deve ser feita pela margem esquerda em formato normal e também pela direita em formato coluna. Os parágrafos devem ser espaçados de ao menos, uma linha em branco. O espaçamento entre colunas pode ser de 3 caracteres se elas forem justificadas pela direita. Em caso contrário este valor passa para 8 caracteres. O comprimento das linhas não pode exceder os 55 caracteres em formato normal e os 35, em formato coluna. O projetista deve evitar de configurar o estilo dos caracteres exclusivamente em caixa alta e o enquadramento pode ser útil para destacar o texto na tela.

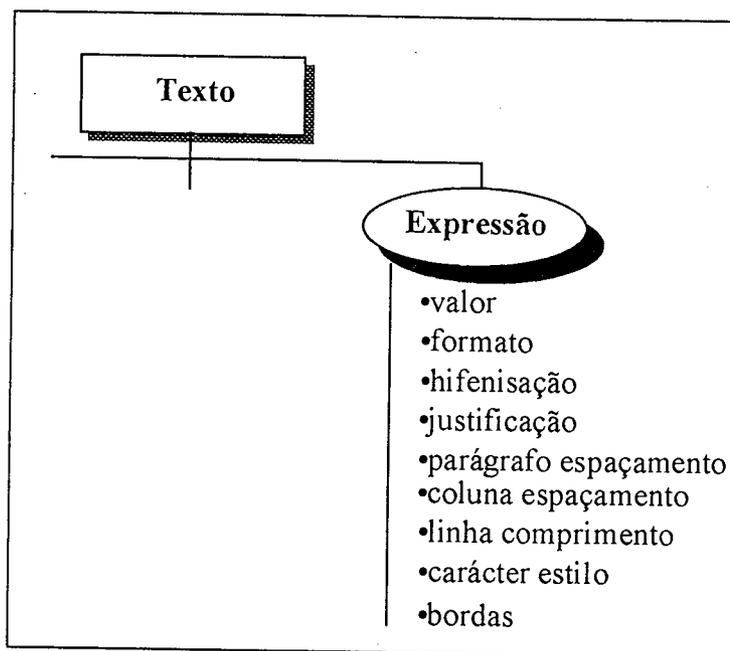


Fig 7.8 - A classe "Texto"

7.1.2.3.2 - A Lista

Uma "Lista" possui recursos para apresentar um conjunto de itens de informação enfileirados verticalmente um em cada linha. O valor do cabeçalho, função identificativa, demanda uma atenção especial. O formato da lista pode ser normal ou indentado, prevendo

assim a possibilidade de hierarquias de itens e sub-itens. O projetista deve prever uma numeração coerente, em algarismos arábicos e suficientemente afastada do texto.

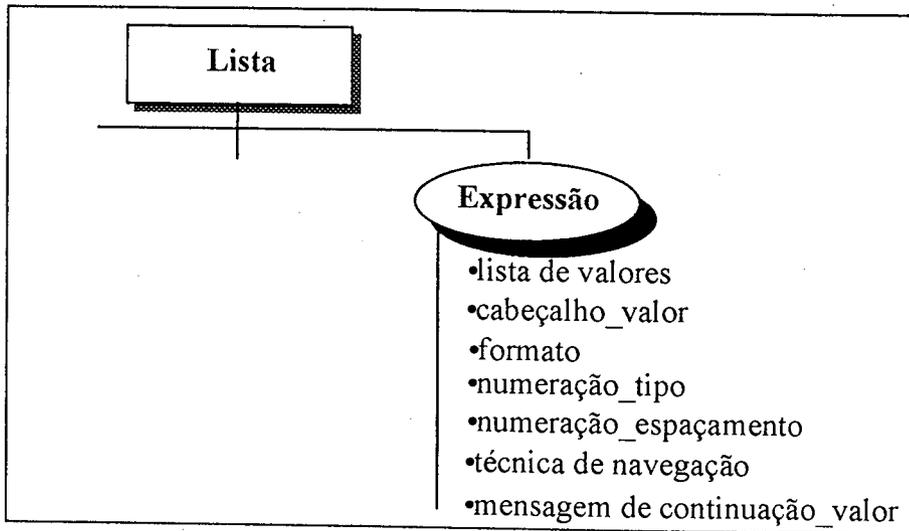


Fig 7.9 - A classe "Lista"

No caso da lista se estender além dos limites de espaço de tela alocado, o projetista pode optar entre dois tipos de navegação possíveis: por paginação ou por rolamento. A paginação se aplica no caso de usuários inexperientes ou quando a tarefa envolver a busca de uma informação isolada. Este recurso demanda a definição referente à mensagem de continuação. Quando o objetivo da tarefa for a busca de relações entre dados, a técnica de rolamento é preferencial.

7.1.2.3.3 - A Tabela

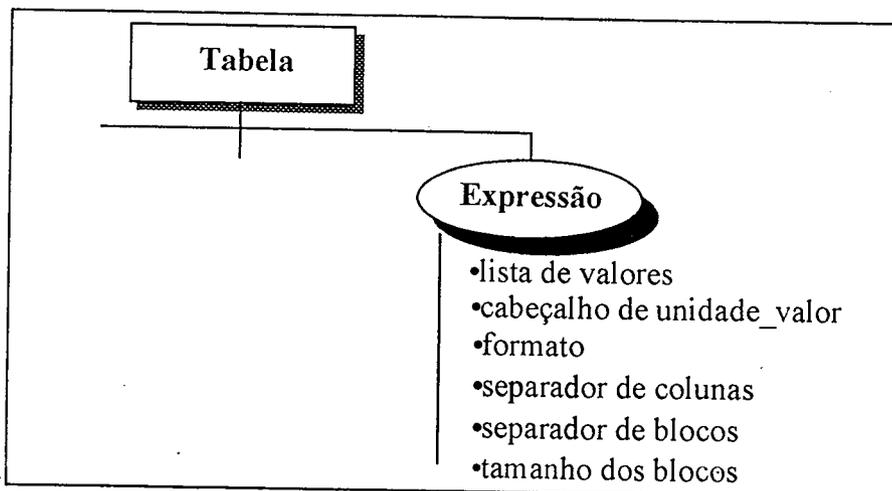


Fig 7.10 - A classe "Tabela"

A tabela herda os recursos da lista e acrescenta os relativos à um cabeçalho descritivo para unidades de medidas além de separadores de bloco e de coluna. O tamanho dos blocos deve girar em torno de 5 linhas.

7.1.2.3.4 - O Gráfico elaborado

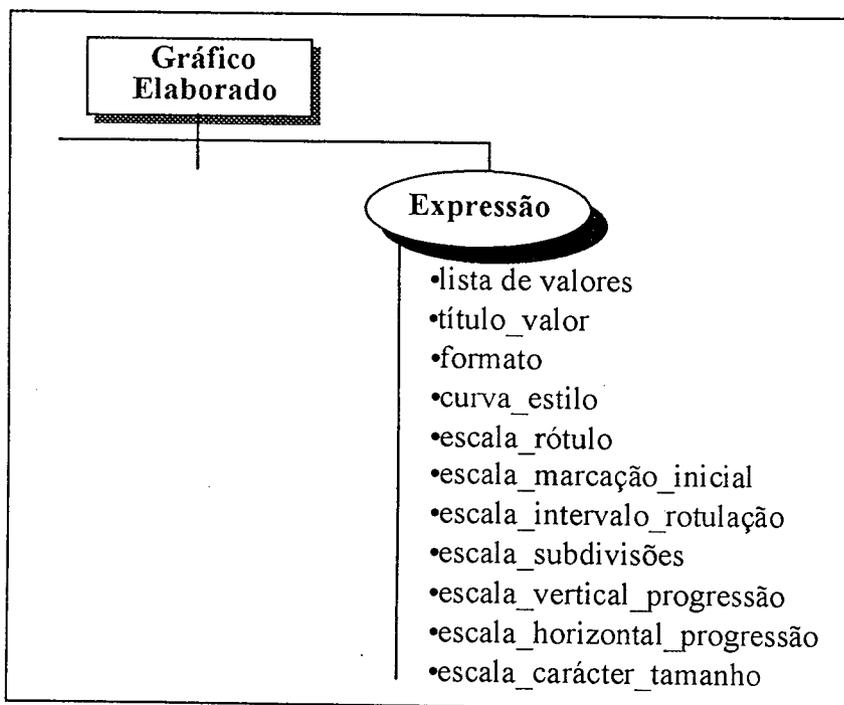


Fig 7.11 - A classe "Gráfico Elaborado"

Os objetos de forma gráfica elaborada possuem recursos para apresentar espacialmente dados ou variáveis correlacionadas. Assim, seus diferentes formatos se aplicam para;

- multi-linha -> análise de tendências,
- gráfico de superfícies -> exame de valores cumulativos,
- gráficos de barra -> exame de amostras à intervalos discretos,
- gráficos de setor -> análise de valores que representam partes de um todo.

Seja qual for o tipo de gráfico, o seu título deve ser descritivo das correlações apresentadas. As curvas podem ser diferenciadas através do uso de traços pontilhados, símbolos geométricos, espessuras e cores. Na configuração das escalas merecem atenção as definições relativas ao rótulo geral, ao tamanho das letras, à marcação inicial (0), ao intervalo

de rotulação (a cada 1,2,5 e 10 marcações), ao número de subdivisões (<9) e às progressões horizontal (esquerda->direita) e vertical (baixo->cima).

7.1.2.3.5 - O Desenho/Diagrama

Um diagrama é uma representação gráfica, que pode ser um desenho ou uma fotografia de um objeto ou de uma estrutura, realizados com o objetivo de mostrar as relações espaciais entre o todo e seus componentes. No caso de limitação de espaço eles podem ser organizados em seções, devendo então ser previstos botões indicadores da visualização das diversas seções. Estes botões podem ter a forma de letras dispostas sobre a vista geral.

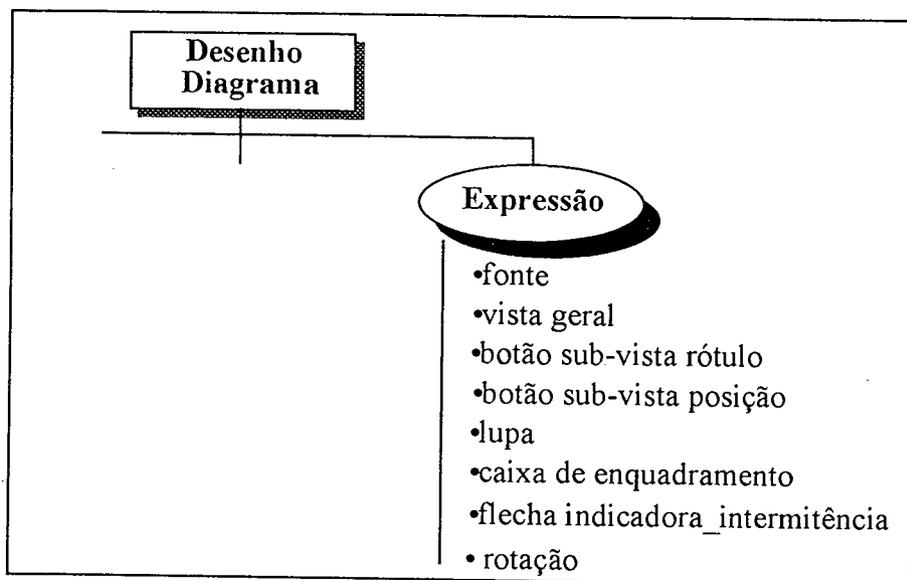


Fig 7.12 - A classe "Desenho/Diagrama"

Uma lupa móvel ou zoom devem estar disponíveis para auxiliar na visualização dos detalhes do diagrama, que pode também contar com o apoio de uma função de rotação. As partes importantes podem ser destacadas através caixas de enquadramento ou de flechas intermitentes.

7.1.2.3.6 - O Diagrama de fluxo

Os diagramas de fluxo são representações gráficas elaboradas para a apresentação esquemática de dados logicamente relacionados em um processo sequencial. Elas se aplicam especialmente para tarefas de resolução de problemas e de gestão de projetos. Seus elementos devem ser apresentados segundo uma ordem lógica, respeitando as convenções estabelecidas em termos de esquerda para a direita, do alto para baixo e no sentido dos ponteiros dos

relógios. As trajetórias mais prováveis devem ser minimizadas, o que pode ser conseguido através da antecipação de pontos de decisões com o menor número de alternativas. Economizar espaço a partir da combinação de decisões, pode confundir o usuário e é algo a ser evitado. A disposição de alternativas em um bloco deve respeitar algum tipo de lógica e ser coerente. Por exemplo o "sim" deve sempre estar à esquerda e o "não" à direita ou abaixo. Finalmente, a escolha dos formatos dos blocos e dos conectores deve ser consistente com o tipo de ação e de relação representados.

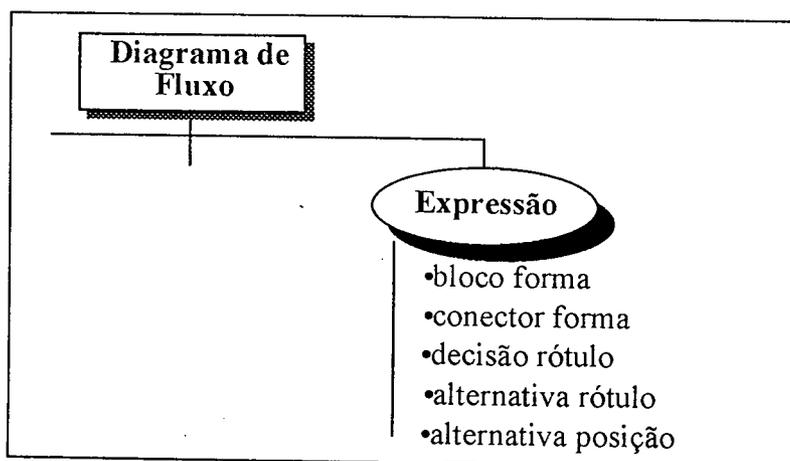


Fig 7.13 - A classe "Diagrama de Fluxo"

7.1.2.3.7 - O Mapa

Um mapa é uma representação reduzida de uma região que é utilizada para a apresentação de dados físicos e geográficos. Sua apresentação pode assumir, segundo as necessidades do projetista, a forma de um mapa ou de uma carta. As cartas são mais apropriadas para a apresentação de dados relativamente estáveis como índices pluviométricos. Os mapas geográficos se aplicam especialmente na apresentação de dados variáveis como a progressão de massas de ar. Eles devem apresentar uma orientação consistente em termos de norte-sul e devem apresentar uma escala precisa e compatível com os dados a serem apresentados. Um efeito de curvatura pode ser definido quando da apresentação de superfícies vastas. Os rótulos descritivos devem ser posicionados cuidadosamente, de modo que não se afastem de sua referência, não encubram outras informações e não venham a causar um congestionamento visual importante. Uma legenda deve ser definida para códigos de textura, cores e de intensidade de cores.

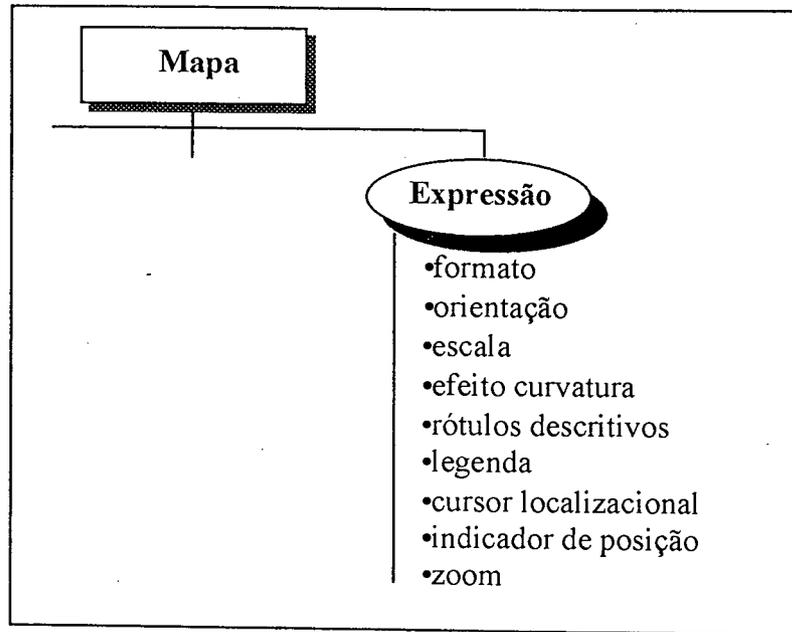


Fig 7.14 - A classe "Mapa"

Norman (Norman, 1991) entretanto considera que o uso frequente de legendas pode atrapalhar o usuário vem a ser uma indicação da falta de adequação nas escolhas feitas em termos dos tipos de representações a adotar. As cores são frequentemente utilizadas para representar densidades e intensidades em mapas geográficos. Entretanto elas constituem escalas substitutivas e valores numéricos pertencem ao domínio aditivo. As cores podem representar dados substitutivos como alinhamentos políticos, tipos de reservas minerais, de florestas, de lavouras, etc. Já a intensidade de cores e texturas, constituem escalas aditivas e se aplicam na representação de dados numéricos como altitudes, demografias, indicadores econômicos, etc. .

Funções de zoom e de navegação devem estar disponíveis aos usuários. Um cursor localizacional e um indicador de distâncias devem ser previstos nos casos de deslocamentos possíveis. Isto com o objetivo de informar, gráfica e numericamente, a posição do usuário sobre a carta.

7.2 - As Formas Auxiliares

A forma auxiliar atua sobre uma forma básica para complementar a carga informacional de uma expressão. Ao modificar uma forma básica, o projetista deve tomar cuidado com o resultado final da expressão, em termos de clareza, legibilidade e tempo de resposta na apresentação.

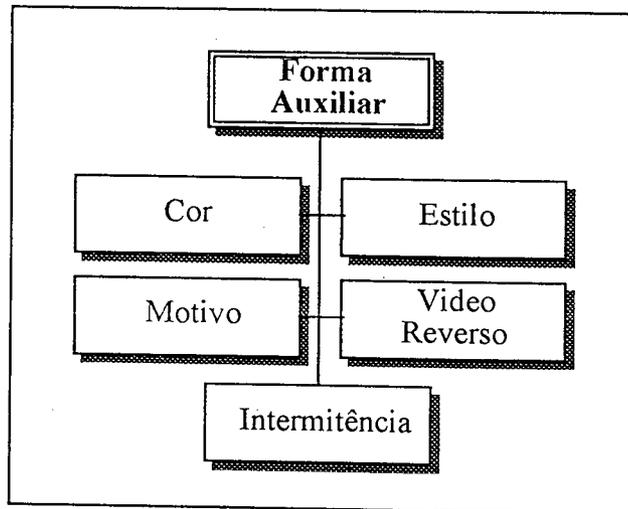


Fig 7.15 - As classes de Formas Auxiliares

7.2.1 - A Cor

O emprego das cores na concepção de interfaces homem-computador tem sido alvo de numerosas recomendações ergonômicas. Elas aconselham o uso de cores para transmitir informações, chamar a atenção, contrastar e associar objetos de interação. O uso puramente decorativo é desaconselhado. Para que sua utilização seja eficaz, deve-se tomar cuidado com três aspectos: (i) a legibilidade final da informação, (ii) com os efeitos das cores sobre a performance cognitiva do usuário e (iii) com as possibilidades dos dispositivos físicos. Estas precauções visam conter a confusão visual resultante do emprego arbitrário e exagerado de cores não pertinentes.

A classe "Cor" propõe recursos para colorir as partes de um OIA^e. Para o fundo e os contornos as cores saturadas, em especial o azul saturado, devem ter a preferência. O verde e o vermelho devem ser evitados para colorir contornos. O azul, o magenta e o rosa devem ser evitados para os caracteres apresentados em primeiro plano.

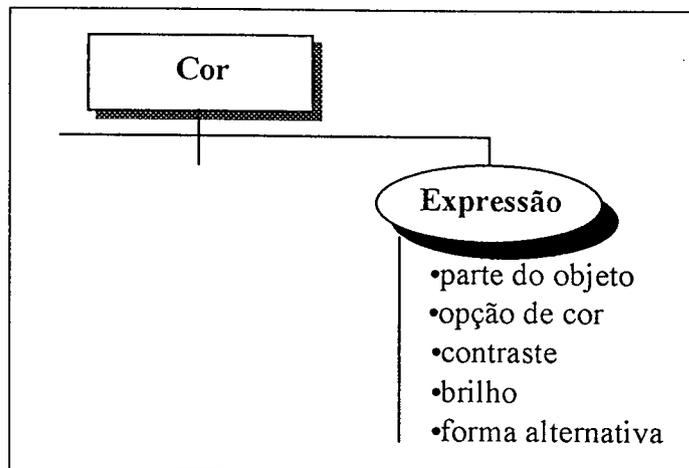


Fig 7.16 - A classe "Cor"

O código de cores deve prever um elenco reduzido e equilibrado de opções (não mais do que 10 ou 11 cores). As cores extremas devem ser descartadas, preferindo-se aquelas centrais, opostas e equidistantes no espectro. Estas não devem estar associadas a mais do que um significado, que deve respeitar os seguintes estereótipos naturais:

- o vermelho deve ser utilizado para perigo, alarme, atenção, alerta, calor e comandos de interrupção.
- o amarelo para advertências, teste e lentidão.
- o verde para passagem livre, normalidade, vegetação e segurança.
- o laranja para valor limite e radiação.
- o azul para frio, água, céu e calma;
- o cinza para inatividade, neutralidade;
- o branco é uma cor neutra.

As variações de cores, definidas a partir da luminosidade e do contraste não podem ser mais do que três e devem respeitar os significados da cor principal. Altos níveis de contraste e de iluminação podem ser definidos para as cores utilizadas no destaque de dados importantes.

Devido à co-habitação de dispositivos físicos coloridos e monocromáticos, a cor não deve ser utilizada como única forma de uma expressão. Assim, os OIA^e desta classe propõem ao projetista a definição de uma forma alternativa e redundante (brilho) para o significado corrente.

7.2.2 - O Estilo

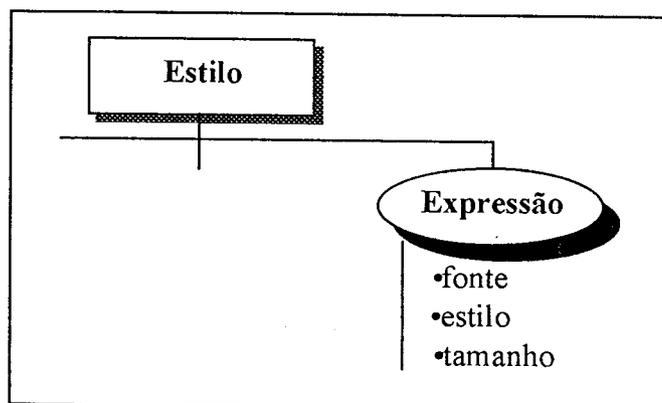


Fig 7.17 - A classe "Estilo"

A classe "Estilo" complementa as capacidades em termos de transmissão de informação da forma tipográfica. Os recursos desta classe incluem fontes, estilos e tamanhos. A utilização das fontes deve respeitar os objetivos definidos para elas. Alguns exemplos são:

Arial -> títulos e cabeçalhos de documentos

Avant Garde -> grandes títulos

Courier -> documentos impressos, cartas padronizadas, correspondência

Helvética -> relatórios, títulos de capítulos, de seções, código de programas

LetterGothic -> texto que deve ser simples e claro

Romano -> correio padronizado

Times -> documentos diversos, de múltiplo uso, comentários em programas

Ultrablack -> etiquetas de embalagens

Os recursos em termos de estilo devem ser usados com parcimônia, para discriminar ou destacar uma informação textual. Eles incluem caixa, negrito, itálico e sublinhado. O tamanho dos caracteres deve ser apropriado (18 à 24 para transparências e 10 à 12 para trabalho normal) e permanecer inalterado na mudança de fontes.

7.2.3 - A Textura

A textura é utilizada como codificação na apresentação de gráficos e mapas. As diferentes opções de textura podem ser empregadas tanto como escalas aditivas como

substitutivas. Se utilizadas juntamente com palavras elas devem ser escolhidas de modo a não prejudicar a leitura.

7.2.4 - O vídeo reverso

O vídeo reverso, ou o contraste é uma codificação binária utilizada para o destaque. A área contrastada deve incluir os espaços vizinhos ao objeto, principalmente se tratando de palavras.

7.2.5 - A Intermitência

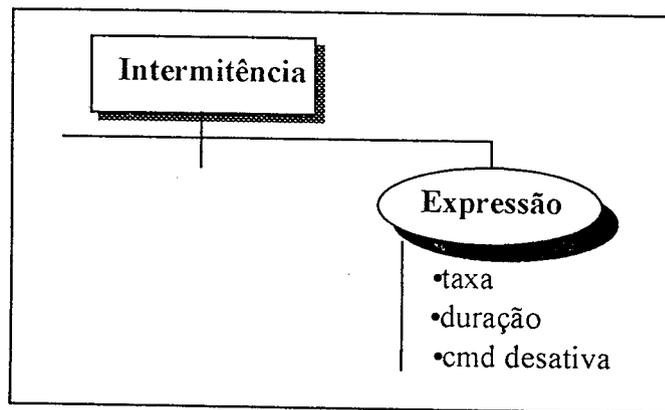


Fig 7.18 - A classe "Intermitência"

A intermitência pode ser aplicada para o destaque em situações excepcionais, em que o perigo de acidentes imponha um caracter de urgência. Esta codificação, essencialmente binária não deve ser aplicada a mais de um elemento de cada vez. A taxa de intermitência pode variar entre 2 e 5 Hz. Na configuração de um OIA^e intermitente, o projetista deve prever um meio, associado a uma tecla, que permita ao usuário desativar a intermitência.

7.3 - Resumo do capítulo

- ✓ As classes de formas estão definidas a partir da relação de subordinação que as envolvem. As formas auxiliares dependem da ocorrência de uma determinada forma básica. Estas últimas são independentes, podendo o projetista escolher livremente entre elas.
- ✓ As formas básicas podem ser visuais ou sonoras.
- ✓ Os sinais sonoros articulam as substâncias de locução, motivos melódicos e o efeitos sonoros.

✓ As formas visuais elementares incluem as gráficos e as tipográficos. As representações gráficas incluem ícones e formas geométricas. As tipográficas incluem, as palavras alfabéticas, numéricas e alfanuméricas.

✓ As formas elaboradas, correspondem a um arranjo de formas elementares e incluem as formas mistas, os textos, as listas, as tabelas, os gráficos, os diagramas, as malhas e os mapas

✓ Uma forma auxiliar atua sobre uma forma básica para complementar a carga informacional de uma expressão. Elas incluem a cor, o estilo, a textura, o vídeo reverso e a intermitência.

8 - A PERSPECTIVA FUNCIONAL

As classes apresentadas até o momento são definidas com base em funções estruturais e formais. As classes de formas têm os recursos para definir e apresentar formas. As classes de estruturas tem os recursos para definir e manter relacionamentos.

Uma outra maneira de definir funções baseia-se na interação de classes de OIA^e com o usuário e com a aplicação. Nela a característica marcante corresponde à capacidade que tem um interador de acionar e de permitir que seja acionado pelo usuário ou pela aplicação. Assim, a combinação destas características define as classes de componentes de uma ferramenta genérica: mostradores, controles e comandos. Reunidos, os exemplares destas classes formam interadores compostos, os painéis de controle.

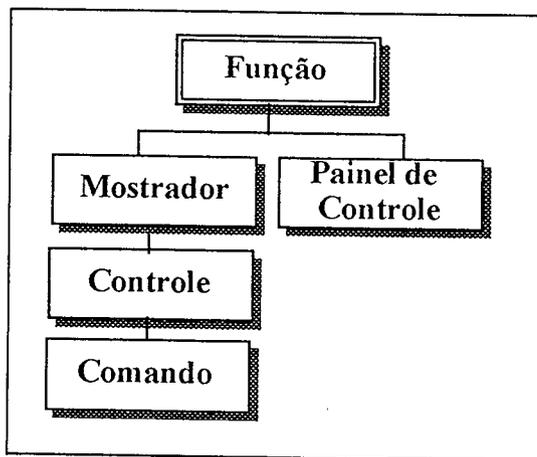


Fig 8.1 - A hierarquia de funções

Os mostradores são acionados pela aplicação para a apresentação de dados e de informações aos usuários. Os controles são mostradores que permitem a interação com o usuário para a recepção de dados. Os comandos são controles que têm os recursos para acionar uma função da aplicação. Os painéis de controle agrupam os mostradores, controles e comandos apropriados a uma ação ou a uma tarefa do usuário.

8.1 - Os Mostradores

Os mostradores são interadores que apresentam a capacidade de "acionar" o usuário através da apresentação das diferentes classes de dados e de informações de que ele necessita para pensar sua tarefa. Por outro lado eles são acionados por um programa aplicativo, que

define quando, como e com quais valores os mostradores devem ser propostos ao usuário. Esta classe de OIA^e corresponde a uma área da tela, ou campo, que pode assumir formas e valores diferentes.

A origem dos valores a serem apresentados na tela distingue os mostradores de dados dos mostradores de informação. Os valores apresentados pelos mostradores de informação são definidos pelos projetistas. Os mostradores de dados apresentam valores provenientes de um contexto, estando fora do controle dos projetistas.

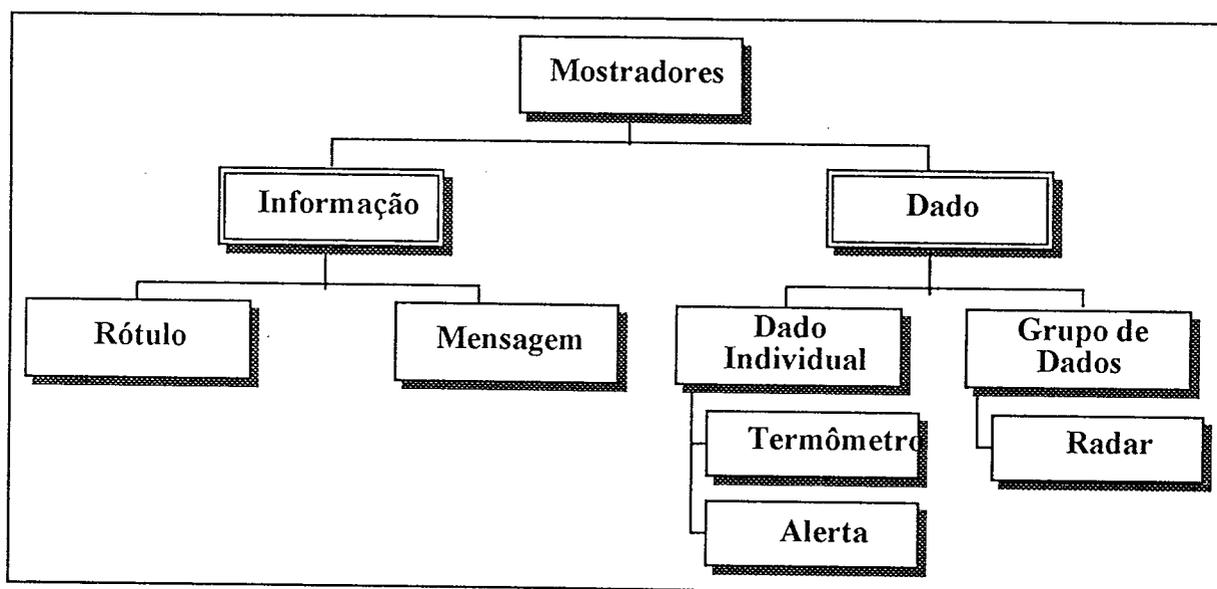


Fig 8.2 - As classes de Mostradores

8.1.1 - O Mostrador de Informação

Uma informação é definida como uma mensagem do projetista ao usuário. Ela desempenha uma função fundamental para as manipulações cognitivas dos usuários envolvendo dados, controles e comandos. A ausência de informação inviabiliza a operação de um sistema.

Os mostradores de informação possuem os recursos para apresentar a mensagem do projetista através das diferentes formas sonoras, visuais gráficas ou tipográficas.

As classes de informação distinguem Rótulos e Mensagem a partir dos recursos de forma que admitem.

8.1.1.1 - O Rótulo

Um rótulo corresponde à um campo que admite somente os recursos de forma elementares. Assim, estes interadores podem assumir a forma de um ícone, de um sinal geométrico, de uma palavra, de um motivo musical ou de um efeito sonoro.

Os rótulos alfabéticos não devem se valer de jargões, devendo ter um palavreado distinto e significativo. Caso demandem uma atenção particular as formas auxiliares podem ser empregadas como expressão discriminativa. O tamanho dos caracteres deve estar definido segundo o critério da legibilidade. A utilização de letras maiúsculas é aconselhável nos rótulos que, conforme sua orientação, devem ser escritos da esquerda para direita e de cima para baixo. O posicionamento relativo ideal é acima ou a esquerda objeto a que fazem referência. Neste particular eles devem estar suficientemente isolados ao mesmo tempo em que permanecem próximos do objeto.

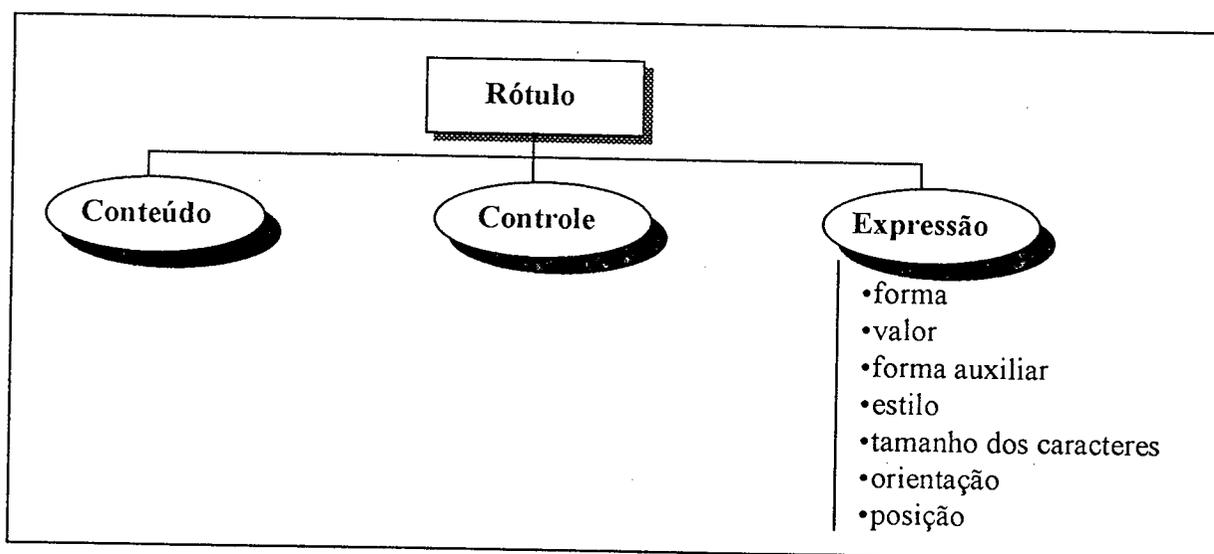


Fig 8.3 - A classe "Rótulo"

Em suas relações com outros sinais, esta classe de OIA^e desempenha diferentes tipos de funções. Estas incluem a identificação, a descrição, a incitação a indicação e a separação. Face a estas funções e às relações com diferentes objetos é possível definir algumas das instâncias de um rótulo.

O título é um rótulo identificativo forçosamente tipográfico. Ele é empregado como função de identificação por diversas classes de objetos: textos, gráficos, tabelas, janelas, caixas de diálogo, etc.

Um cabeçalho é um rótulo que é gerado na especificação de uma lista ou tabela. Ele deve ser destacado ou discriminado através do tamanho ou do estilo de caracteres sublinhados, de cores e até mesmo através do brilho (vídeo reverso)

Da mesma forma que o cabeçalho se define na relação com listas, um prompt se define na relação com campos de dados. Um prompt é o rótulo de incitação, gerado na especificação de um campo de dados. Ele deve ser destacado ou diferenciado através de um posicionamento adequado tela.

Os marcadores são rótulos indicativos gerados na especificação de objetos como itens de seleção, opções de menu, botões de controle e botões de comando. Ele só admite a forma de sinal simbólico

O separador é um rótulo que se define na relação com listas. Este só admite a forma geral de uma linha, que pode ser cheia, tracejada, pontilhada, etc....

8.1.1.1 - A Mensagem

A mensagem define a classe de informação que admite as formas elaboradas de texto, tabela, lista, gráfico, diagrama, mapa e locução sonora.

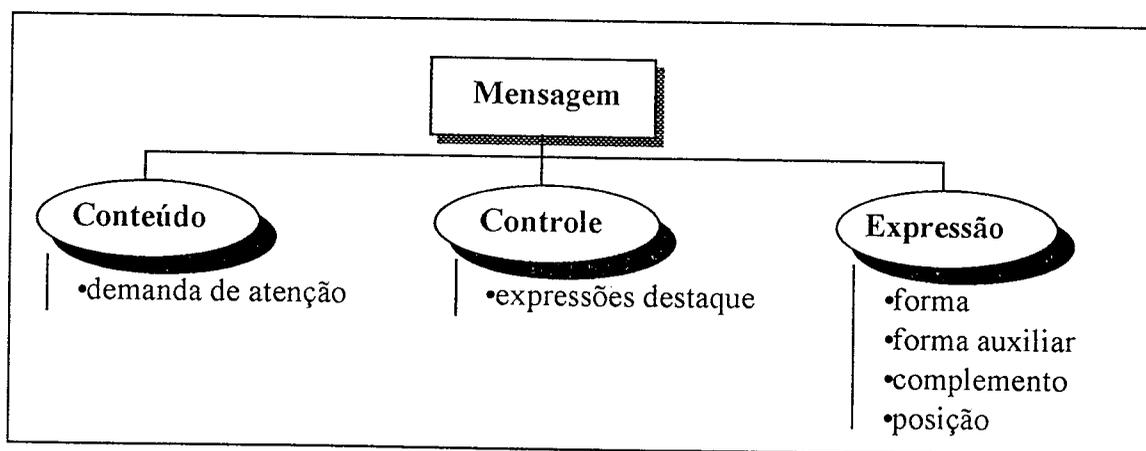


Fig 8.4 - A classe "Mensagem"

A forma textual é especialmente adequada para lidar com noções abstratas e expressar deduções lógicas. Este tipo de mensagem deve apresentar frases afirmativas e diretas, na voz ativa, evitando pontuações desnecessárias e apresentando os argumentos segundo uma ordem lógica. Os exemplares desta classe proporcionam ao projetista a configuração de expressões destaque, a serem ativadas em casos de urgência. Para garantir a transmissão da informação as

mensagens textuais propõem a participação de um complemento com uma forma de expressão alternativa, por exemplo um diagrama ou um gráfico.

8.1.2 - O Mostrador de Dados

Os valores ligados às evoluções do sistema usuário-tarefa-dispositivo informatizado não podem ser definidos estaticamente ou de antemão pelos projetistas. Eles correspondem a variáveis do contexto e para poder apresentá-los ao usuário os "mostradores de dados" lançam mão de recursos estruturais de correlação. Todo mostrador de dados possui uma estrutura de correlação, que pressupõe a definição de uma fonte de valor e de uma equação relacionando este valor aos parâmetros descritivos de uma determinada forma. A construção de representações icônicas dinâmicas valem-se deste mesmo princípio (Fairchild, Meredith, & Wexelblat, 1989; Gray, Waite, & Draper, 1990), bastante pertinente sob o ponto de vista da modularidade dos sistemas interativos.

Sua tipologia diferencia os mostradores de dados individuais dos mostradores de grupos de dados. Os objetos do primeiro tipo estão associados a somente uma variável do contexto, os de segundo representam uma lista destas variáveis.

8.1.2.1 - O Mostrador de Dados Individual

Sob o ponto de vista ergonômico, os mostradores de dados individuais devem alocar recursos de rótulos identificativos e de rótulos descritivos para explicitar as unidades dos valores apresentados. Uma forma tipográfica deve ser definida quando houver uma necessidade de precisão de leitura do dado. No caso de valores que variem rapidamente, uma forma gráfica analógica deve ser definida. A forma auditiva pode ser uma alternativa nos casos de telas saturadas ou de postos de trabalho móveis. O formato, ou o perfil dos dados alfanuméricos merece uma atenção especial. Sempre que for possível o projetista deve adotar valores padronizados e coerentes com as convenções de usuário. Se o formato resultar demasiadamente longo ele deve ser dividido em grupos lógicos. No que se refere às formas auxiliares, as cores, em especial o amarelo, pode ser empregado para agrupar dados espalhados na tela. Cores mais saturadas (contraste) ou mais intensas (brilho) podem ser utilizadas para o destaque de dados críticos. A intermitência deve ser usada para o destaque, com o cuidado de adotar codificações binárias. Ela não deve ser usada no caso de dados alfanuméricos. O

tamanho dos caracteres também pode ser utilizado como forma de destacar dados urgentes. No posicionamento de dados, a principal recomendação está ligada à sua consistência.

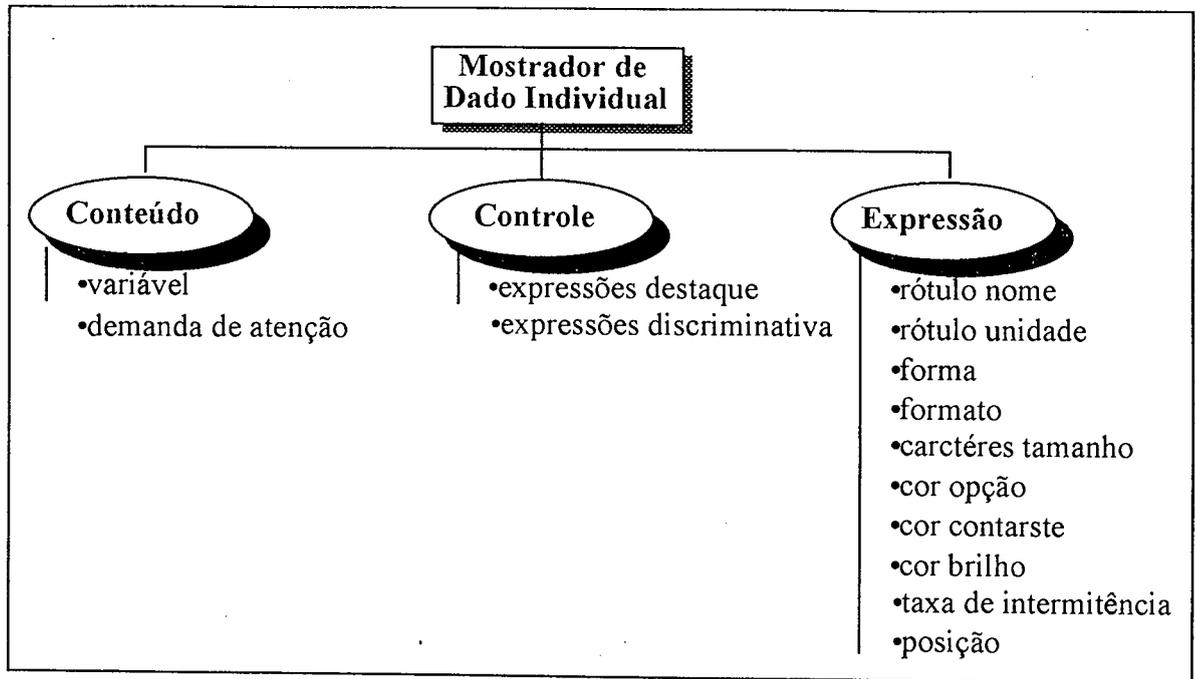


Fig 8.5 - A classe "Mostrador de Dado Individual"

8.1.2.2 - O Mostrador de grupos de dados.

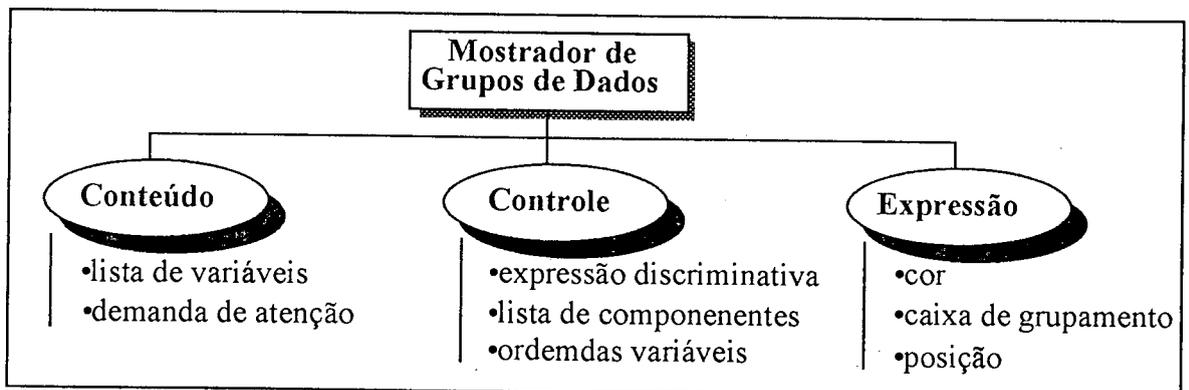


Fig 8.6 - A classe "Mostrador de Grupos de Dados"

O mostrador de grupo de dados está ligado a uma série de valores de contexto que são apresentados tipicamente através de gráficos, tabelas ou listas. Estas formas facilitam o exame da dados numerosos. A lista de valores do contexto a agrupar são declarados no atributo de conteúdo "lista de variáveis". Esta definição deve seguir os critérios de sequencialidade, de funcionalidade, de importância e de frequência de utilização dos dados. A discriminação de um

grupo de dados pode ser feita através de seu posicionamento, pelo emprego de cores ou pela definição de uma caixa de grupamento.

8.1.2.3 - O Radar, o termômetro e o alerta.

Os exemplares de OIA^e mostradores de dados individuais ou de grupos de dados identificados nas recomendações ergonômicas referem-se às classes de radares, termômetros e alertas.

Os radares são mostradores de grupos que associam listas de variáveis como posição, altitude, velocidade e direção de um conjunto de aeronaves que compartilham de um espaço aéreo. Estas classes possuem a forma elaborada de uma superfície sobre a qual se distribuem os marcadores de aeronaves. O comportamento dos alvos é definido por uma relação de constrangimento. Os projetistas devem tomar cuidado com a maneira de destacar ou distinguir estes alvos, evitando a sobrecarga visual da tela.

Já o termômetro é um mostrador individual que associa temperaturas às características de uma forma. Esta pode ser análoga ou digital, sendo aconselhado o emprego das cores azul, para temperaturas reduzidas e vermelho para as elevadas.

Um "alerta" destina-se à apresentação de variáveis ligadas ao controle de um processo.

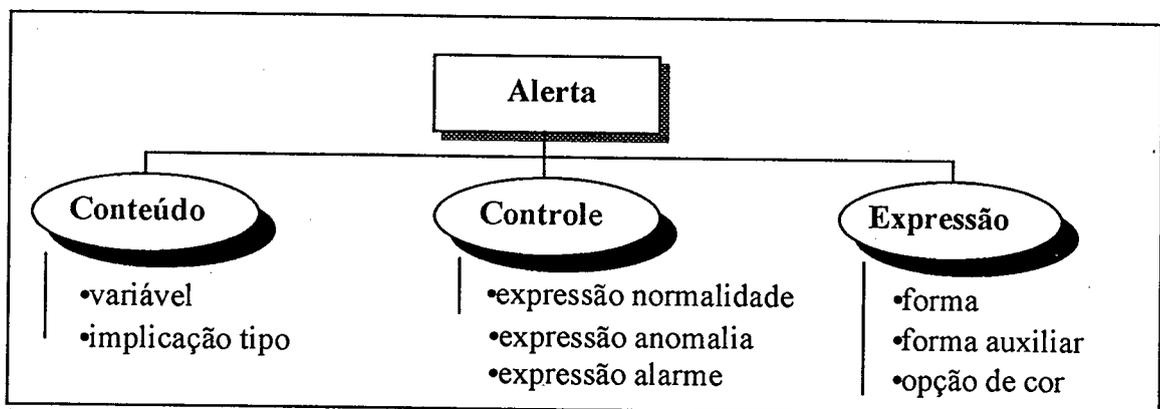


Fig 8.7 - A classe "Alerta"

Ela proporciona aos projetistas a possibilidade de declarar expressões associadas ao atributo de conteúdo "tipo_de_implicação". Caso este objeto esteja implicado em uma situação de alarme ou de anomalia do sistema, seu controle solicita a ativação da expressão de anomalia ou da expressão de alarme que vai sinalizar sua ocorrência ao usuário. Uma expressão para

signalizar normalidade também está prevista. Conforme os estereótipos naturais, a cor verde deve ser empregada para a expressão de normalidade e o rosa para a expressão de anomalia. A intermitência pode ser utilizada para a expressão de alarme.

Aqui o aspecto interessante está no elevado nível de abstração em que ocorrem as comunicações entre objetos durante o funcionamento do sistema. A aplicação envia ao mostrador "alerta" uma mensagem informando simplesmente "normalidade", "anomalia" ou "alarme". O mostrador faz o resto, adotando sua expressão à nova realidade. A comunicação a um baixo nível de abstração se daria nos termos de "set color to green", "set color to pink" ou "turn on blink". Este tipo de mensagem não é absolutamente significativo do que está ocorrendo e certamente contribui para aumentar os gastos de manutenção do sistema.

8.2 - Os Controles

Os controles são OIA^e sensíveis às ações do usuário, lhe proporcionando facilidades em termos de edição, seleção e manipulação direta. Esta classe de objetos não tem os recursos para acionar uma funcionalidade da aplicação. Assim, ela só possibilita a entrada de dados, os quais definem os parâmetros para a ativação posterior de um comando.

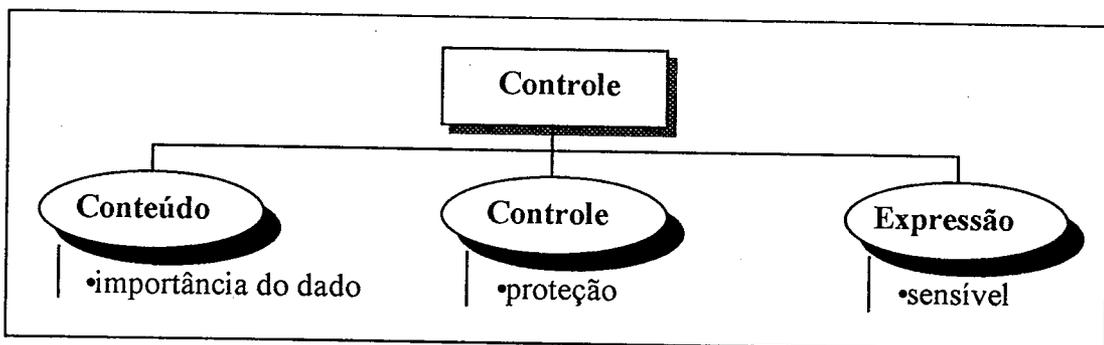


Fig 8.8 - A classe "Controle"

Em resposta às recomendações ergonômicas, os objetos desta classe possuem um mecanismo de proteção aos dados que lhe estão associados. No momento em que o valor de conteúdo "Importância dos dados" for instanciado como "essencial", o controle de um objeto passa a não mais interagir com o usuário. O objetivo é de manter a integridade dos valores correntes.

Os controles podem ser de três tipos; os editores; os seletores e os manipuláveis.

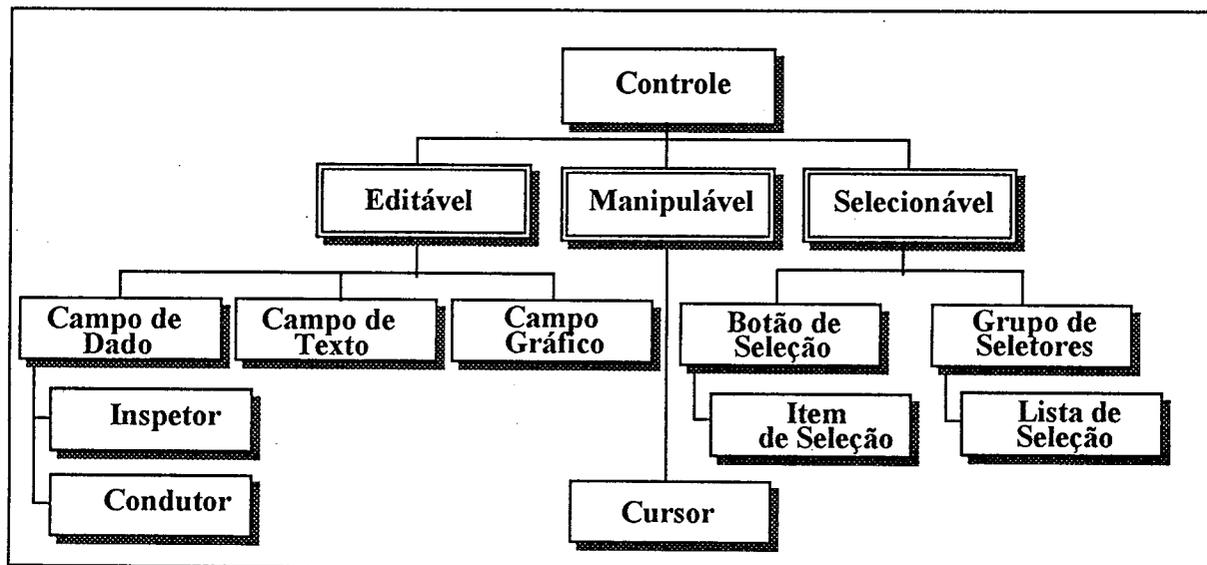


Fig 8.9 - As classes de Controle

8.2.1 - O controle manipulável - o cursor

Os cursores representam a classe de controles verdadeiramente manipuláveis do modelo. Seu objetivo é permitir a designação ou a indicação de uma posição na tela. É através do cursor do mouse que o usuário seleciona e aciona parâmetros e funcionalidades dos diferentes objetos.

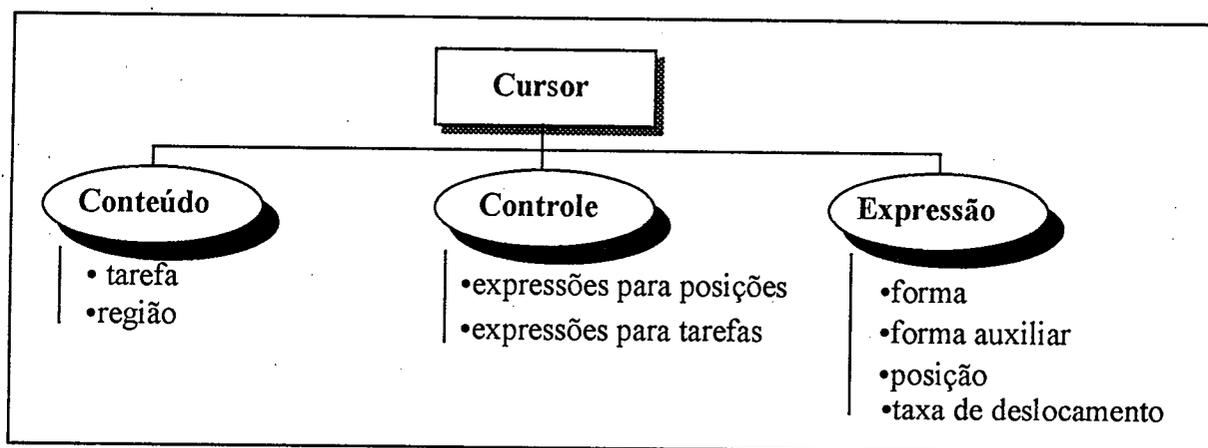


Fig 8.10 - A classe "Cursor"

Sua forma deve ser diferenciável, ao mesmo tempo em que não prejudique a visualização dos objetos que encobre. O cursor do mouse oferece a possibilidade de definição de diversas formas que se alternam em função de seu posicionamento ou do tipo de tarefa. Neste caso elas devem ser significativas e ao mesmo tempo fornecer um apoio ao usuário. Por

exemplo quando a tarefa exigir precisão na indicação, a forma definida pode incluir um círculo. O projetista deve definir valores para a taxa de deslocamento levando em consideração um compromisso entre a velocidade e a precisão no posicionamento.

8.2.2 - Os controles editáveis

As possibilidades em termos de controles editáveis incluem os campos para a edição de palavras, de textos e de objetos gráficos.

8.2.2.1 - O campo de dados

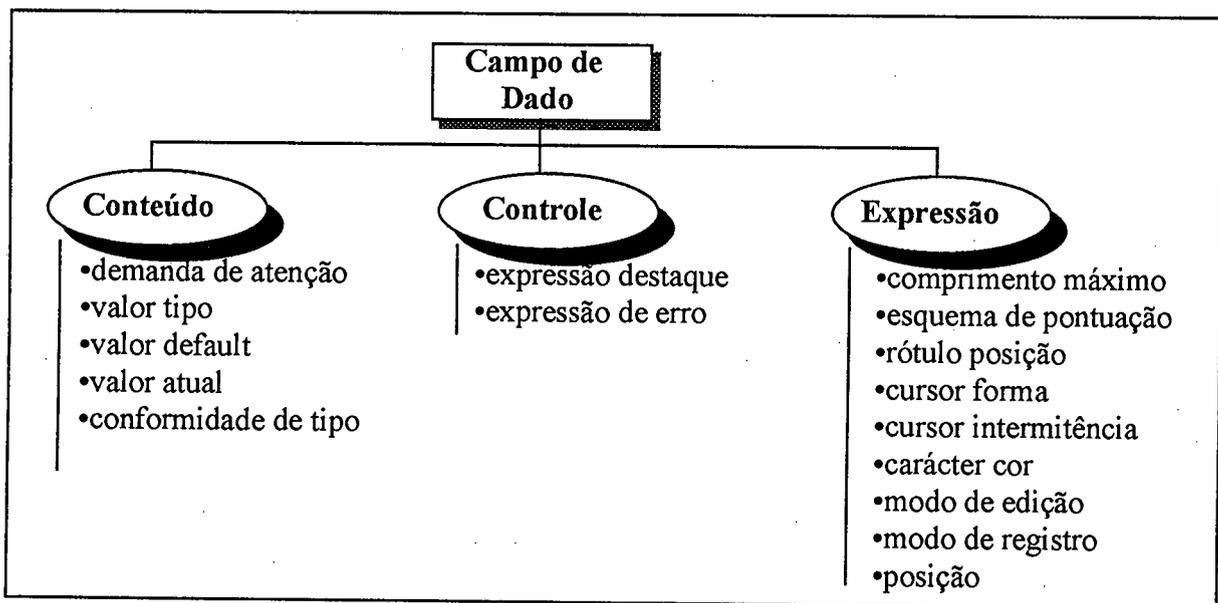


Fig 8.11 - A classe "Campo de dado"

Um OIA^e da classe "Campo de Dados" proporciona facilidades para edições uni-lineares. Eles recebem dados cujos valores não podem ser previstos pelo projetista e cujos comprimentos não excedam os 40 caracteres. Um esquema de pontuação com espaços, vírgulas, hífen ou barras pode ser definido para dividir entradas com mais de 10 letras ou 5 dígitos em grupos de 3 a 4 caracteres. Com o objetivo de minimizar as ações do usuário o projetista pode especificar um valor a ser proposto "par défaut". A escolha de um modo de edição "insert/delete" deve ter a preferência sobre o modo "typeover". Mas seja qual for a definição ela deve ser mantida consistente durante o projeto da interface. Os métodos para o registro dos valores entrados podem considerar equivalentes as letras maiúsculas e minúsculas. O preenchimento dos zeros e dos pontos decimais desprezados pelo usuário também deve ser considerado. Os dados

entrados devem ser checados quanto à sua conformidade de tipo e cabe ao projetista declarar uma expressão de destaque para assinalar ao usuário quando os erros acontecerem.

O destaque de um campo de dados pode se dar pelo seu posicionamento ou pelo emprego das cores. Todo o campo de dado deve apresentar um rótulo identificativo ou convidativo (prompt), que deve ser posicionado de uma maneira consistente. Em geral ele deve ser colocado à esquerda do campo e alinhado pela sua margem inferior. O cursor deve ser discriminado através de uma forma de caixa ou de diamante ou através da intermitência. Recomenda-se que a cor dos caracteres digitados seja a turquesa.

8.2.2.1.1 - Os campos condutor e inspetor

Algumas recomendações ergonômicas referem-se à aspectos específicos dos campos de dados que podem ser definidos como recursos de duas especializações suas: os campos inspetor e condutor.

O inspetor é proposto como um detalhamento do exame de conformidade realizado pelo campo de dado. Um campo inspetor fornece recursos para determinar e reportar ao usuário quando o valor entrado está ok, escapa à normalidade ou desvia da tolerância. As cores podem ser utilizadas na definição das expressões "desvio" e "fora do usual". A expressão normalidade pode explorar o estereótipo da cor verde

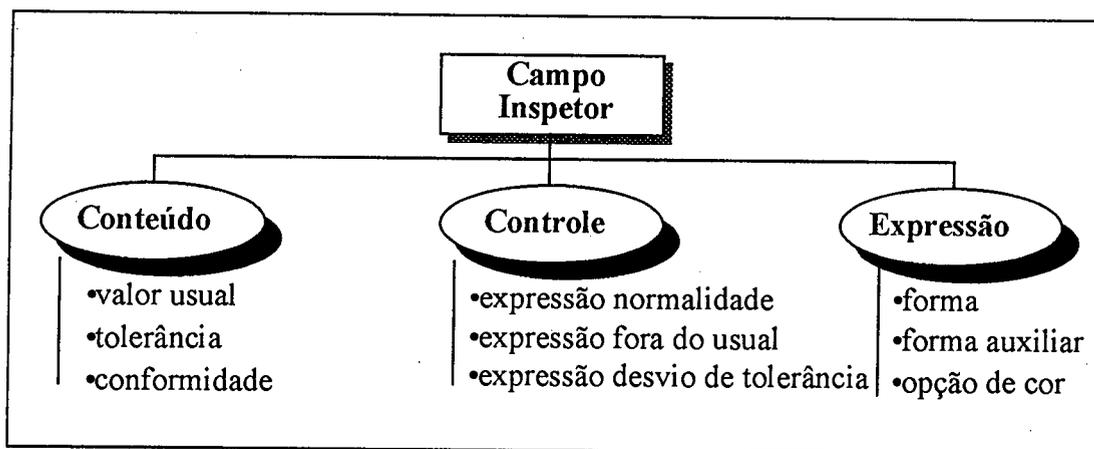


Fig 8.12 - A classe "Campo Inspetor"

O campo condutor é definido de modo a auxiliar nas ações de atualização de dados. Este campo define os recursos para avisar ao usuário sobre o momento propício para a atualização de um valor. O mecanismo se baseia no valor da variável de conteúdo "atualização". No momento em que este valor se tornar verdadeiro, o controle solicita o

acionamento de uma expressão específica e predefinida pelo projetista. Sugere-se o emprego da cor amarela na definição da "expressão atualização".

Através da definição destas duas classes é possível exemplificar o efeito de repartição da semântica entre os componentes de um sistema interativo, que é proporcionado pela arquitetura PAC (Coutaz, 1990). Ele se verifica na delegação de uma parte das competências do núcleo funcional aos objetos de interação, com a vantagem de diminuir a frequência de trocas entre eles. Estes dois tipos de campos reivindicam para si a tarefa de inspecionar e de conduzir. Eles definem um nível de competência usualmente superior à dos objetos de interação. Este efeito de repartição semântica dá margem à formação do que se pode chamar de semântica da interação. Ela resulta da reunião de tarefas que deixam o núcleo funcional, sendo atraídas pelos OIA^e.

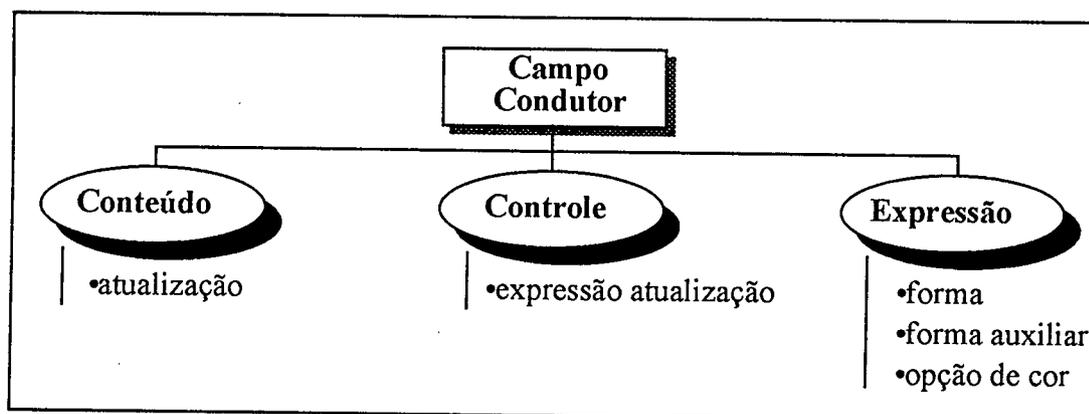


Fig 8.13 - A classe "Campo Condutor"

8.2.2.2 - O Campo de Texto

O campo de texto apresenta os recursos de edição multi-linha para receber do usuário dados na forma textual. Seu tamanho em termos do número e do comprimento de linhas deve ser adequado para proporcionar um desempenho eficiente na tarefa de entrada de textos. Para a facilidade de leitura o comprimento das linhas não deve exceder os 40 caracteres. Sob o ponto de vista de edição, o projetista pode especificar um cursor intermitente para auxiliar na localização do ponto de inserção de texto.

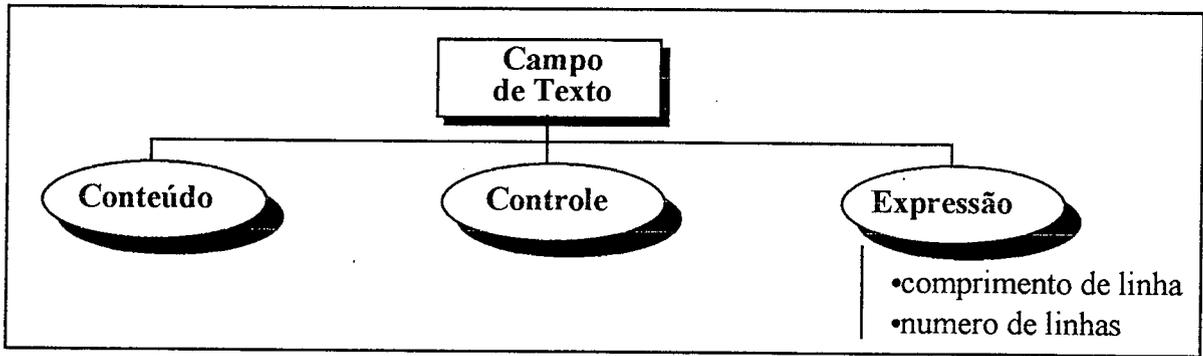


Fig 8.14 - A classe "Campo de Texto"

8.2.2.4 - O Campo Gráfico

O campo gráfico se define nos recursos de edição gráfica que proporciona ao usuário. As recomendações ergonômicas são econômicas em relação à esta classe de objetos. Eles se referem às diferentes formas do cursor, como "+" ou "†", face aos diferentes tipos de tarefa. O centro de rotação de um objeto deve coincidir com o centro do próprio objeto e uma função de zoom deve ser prevista.

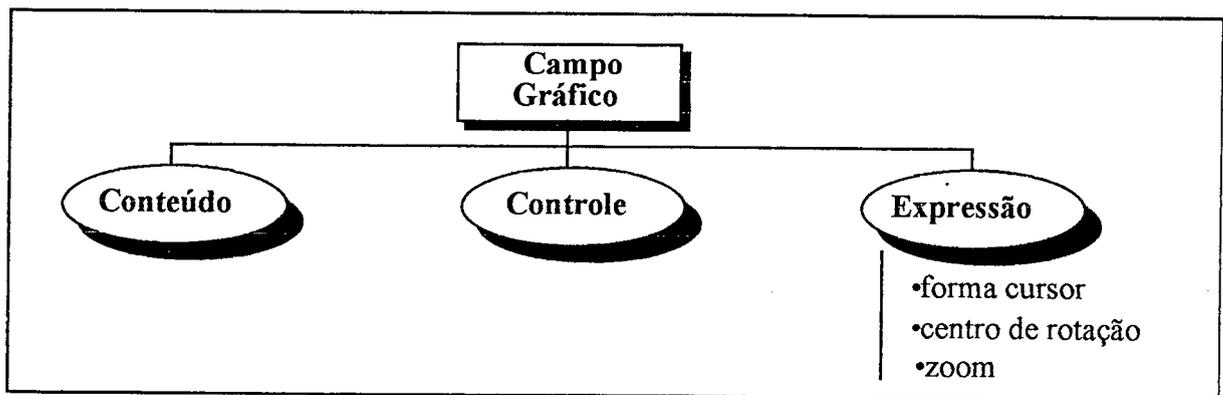


Fig 8.15 - A classe "Campo Gráfico"

8.2.3 - Os controles selecionáveis

Os seletores ou campos de seleção, permitem a entrada de dados através da indicação dos valores predefinidos que neles residem.

8.2.3.1 - O botão de seleção

O botão de seleção corresponde a um campo de seleção que pode assumir três estados: um estado "ativo", onde a seleção é possível; um estado "inativo", onde a seleção não é

possível e finalmente o estado "definido", onde a seleção já ocorreu. Estes devem ser representados através de formas coerentes e suficientemente distintas.

Um botão de seleção apresenta o seguinte comportamento imediato. No estado "inativo" ele não pode ser selecionado. Se selecionado a partir de seu estado "ativo", ele assume o estado "definido". Se selecionado a partir do estado "definido" ele volta ao estado normal "ativo". O comportamento em um grupo define os dois tipos de botões de seleção; o não exclusivo (toggle) e o exclusivo (radio). Em um grupo de botões exclusivos, só um deles pode estar definido ao mesmo tempo. A seleção de um outro botão causa o "desligamento" do anterior. Em listas de botões não exclusivos, diversas opções podem estar definidas simultaneamente. Um botão deste tipo pode ser selecionado livremente sem consequências sobre os outros componentes do grupo.

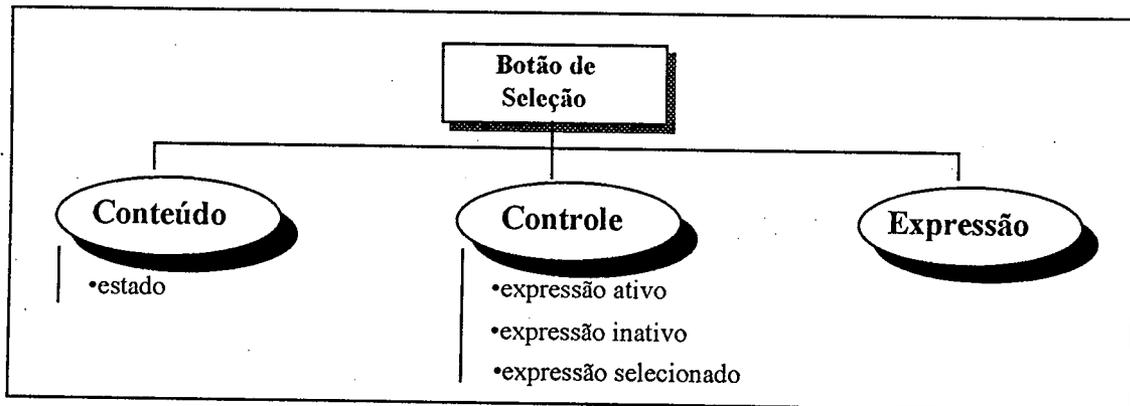


Fig 8.16 - A classe "Botão de Seleção"

8.2.3.2 - O item de seleção

O item de seleção corresponde a um campo para a seleção de um dado de forma elementar gráfico, elementar tipográfico ou misto. O item de seleção invariavelmente integra uma lista de seleção.

Ele pode assumir os estados "inativo", "ativo", em "foco" e "selecionado", para os quais deve ser previstas formas de expressão coerentes e diferenciáveis. No estado ativo o item se encontra sensível a uma ação de seleção. No estado inativo a seleção não é possível. O estado "em foco" corresponde ao momento anterior a uma seleção, quando o usuário posiciona o cursor sobre o item e prepara-se para selecioná-lo. A expressão de "feedback" de seleção informa que uma seleção acaba de ocorrer. A expressão "selecionado" indica ao usuário que aquele item já se encontra definido.

O vídeo reverso pode ser empregado como expressão "feedback". Marcadores, com a forma de asteriscos, flechas ou símbolos especiais podem ser utilizados para as expressões "selecionado". A expressão "inativo" é tipicamente representada pelo tom acinzentado de suas formas.

Um item de seleção apresenta o comportamento básico de um botão, podendo ser exclusivo (radio) ou não exclusivo (toggle).

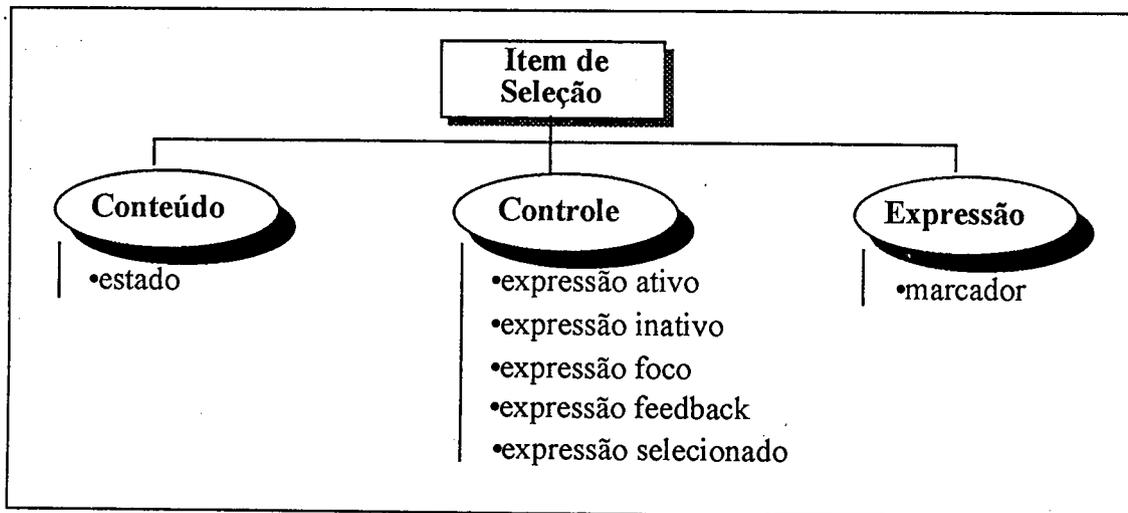


Fig 8.17 - A classe "Item de Seleção"

8.2.3.3 - O grupo de Seletores

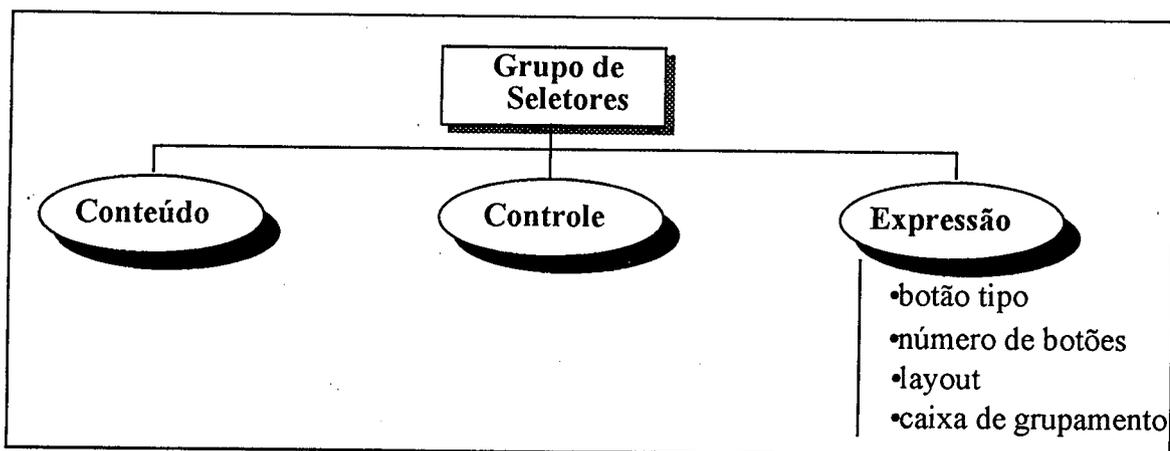


Fig 8.18 - A classe "Grupo de Seletores"

O grupo de botões reúne um máximo de sete botões de seleção ou de rádio que devem estar distribuídos equidistantemente. A caixa de grupamento só deve ser utilizada se eles foram mais do que quatro botões.

Um grupo de botões a ser utilizado para a entrada de dados deve prever botões de rádio quando o conjunto de valores possíveis para um dado forem conhecidos, não excederem sete alternativas e forem mutuamente exclusivos. Se a escolha for múltipla o grupo deve prever botões de seleção.

8.2.3.4 - A lista de seleção

Uma lista de seleção corresponde a um campo de seleção multi-linear que abriga itens tipográficos, gráficos e mistos para uma escolha simples ou múltipla. Elas são empregadas para a entrada de dados cujos valores possíveis sejam conhecidos e em seu conjunto excedam as oito alternativas. O tamanho usual de uma lista deve permitir a visualização imediata de 7+- 2 itens. O projetista deve especificar a ativação de mecanismos de navegação internos (barras de rolamento) quando o número de escolhas possíveis se torne elevado. O limite máximo é de 50 itens que devem ser ordenados, iniciando-se pelas alternativas mais prováveis. Os separadores devem ser empregados para marcar a organização dos itens segundo grupos lógicos. Se qualquer organização não for possível, os separadores devem ser dispostos a cada 6 linhas de alternativas. Eles podem ser definidos como linhas em branco, de traços cheios ou pontilhados.

No caso de restrições de espaço elas podem ser configuradas como listas desdobráveis. Desta forma somente a primeira linha da lista é apresentada inicialmente. Para visualizar as outras alternativas o usuário deve acionar o botão de desdobramento que irá apresentar o restante da lista em modo de sobre-apresentação (primeiro plano da tela). O comprimento das linhas pode ser determinado pelo item mais longo se ele for menor do que 20 caracteres. Caso contrário ele deve ser definido pela média dos comprimentos de cada item.

Assim como para os botões, os itens desativados ou selecionados devem possuir uma forma coerente e diferenciável. A desativação pode ser representada através da cor cinza dos caracteres. Para os itens em foco o padrão é o vídeo reverso. O emprego de pequenos símbolos (✓, * ou flechas) pode ser especificado para marcar os itens já selecionados nas listas com múltiplas escolhas. Toda a lista de seleção deve possuir um rótulo identificativo a ser posicionado acima dela e alinhado pela esquerda. Se a lista for de desdobrar este rótulo deve permanecer à esquerda e alinhado pela sua margem inferior.

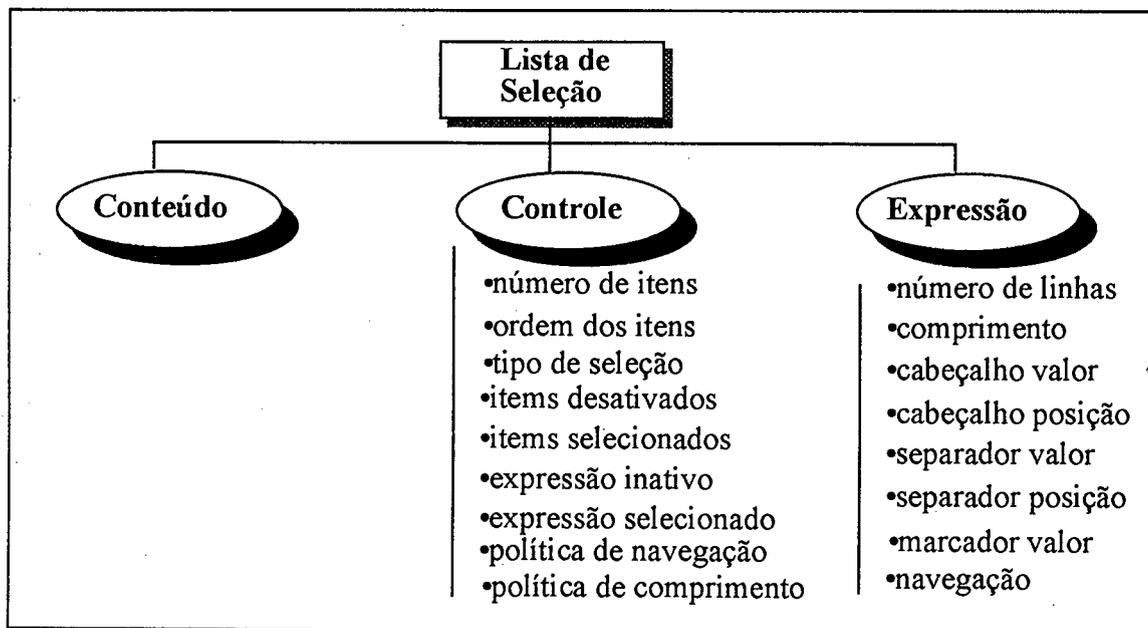


Fig 8.19 - A classe "Lista de Seleção"

8.3 - Os Comandos

A classe de Comandos é uma especialização da classe de Controles cujos elementos abrigam chamadas para as ações da aplicação. Em consequência deste parentesco direto com os controles, as classes de comandos são definidas segundo seus recursos de edição, seleção e de manipulação.

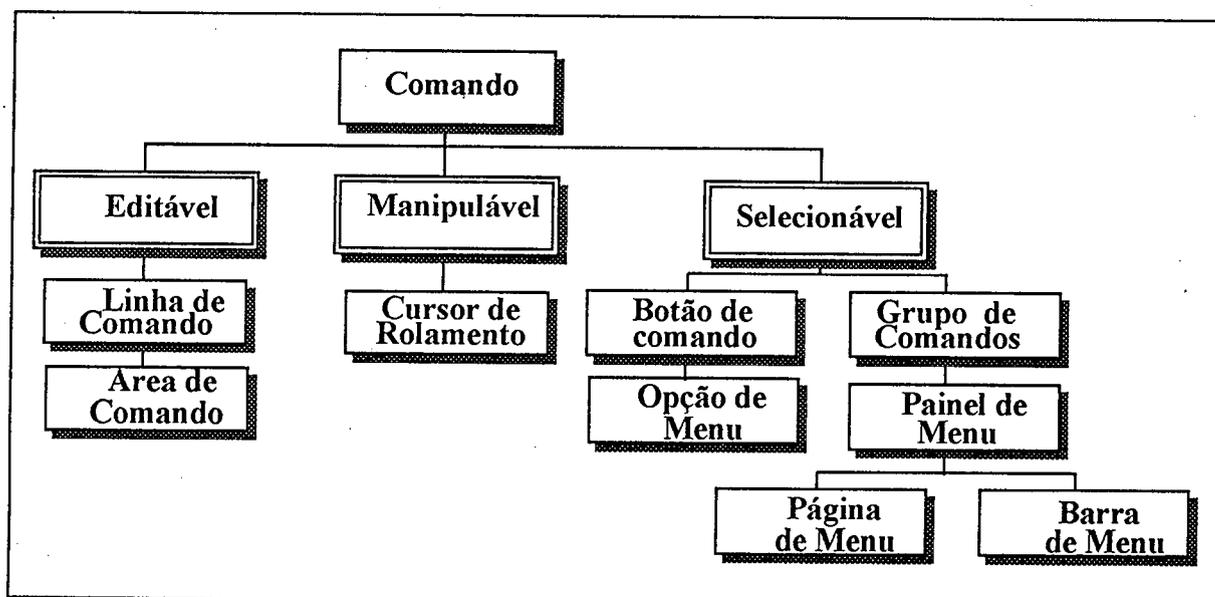


Fig 8.20 - As classes de Comandos

Os exemplos disponíveis em termos de comandos manipuláveis correspondem ao "cursor de rolamento" de uma barra de rolamento e às "marcas" de tabulação e margens de uma régua de edição de textos. O usuário age sobre o cursor para deslocar um documento sob uma janela. Sobre as marcas ele define diretamente as posições de tabulação e das margens de um parágrafo. As recomendações ergonômicas não tecem maiores comentários sobre a utilização deste tipo de comando.

8.3.1 - Os comandos editáveis: linha e área de comando

As áreas e linhas de comando correspondem a campos de edição uni e multi-lineares cujo conteúdo aciona uma funcionalidade do aplicativo. Neste sentido eles proporcionam grande flexibilidade ao usuário que pode se valer de todas as opções de vocabulário previstas numa linguagem de comando. A área de comando em especial fornece recursos de histórico para que o usuário possa avaliar e retomar estratégias de interação. Os comandos deste tipo devem estar localizados na parte inferior da tela e a área de comando não deve ser inferior à quatro linhas.

8.3.2 - Os comandos selecionáveis

Em relação aos comandos editáveis, os selecionáveis facilitam consideravelmente a tarefa do usuário, que realiza uma atividade mental de reconhecimento e não de recuperação. Nesta última o seu rendimento é muito superior. Outro aspecto importante dos comandos selecionáveis está ligado a abstração dos detalhes da sintaxe dos comandos. O usuário é conduzido na entrada de parâmetros de acordo com a opção de comando selecionada e os parâmetros já entrados. Este fato proporciona uma redução importante de erros de sintaxe.

8.3.2.1 - O botão de comando

O botão de comando é um campo para a seleção de comandos. Seus estados possíveis incluem "ativo", "inativo", "armado" e "ativado". O estado "armado" se instala quando o botão é definido como opção "par défaut" em um grupo de comandos. Nesta situação seu acionamento pode se dar tanto pelo mouse como através das teclas "Enter" ou "Return". Os nomes de botões que acionam caixas de diálogo devem ser seguidos de pontos de continuação

"...".

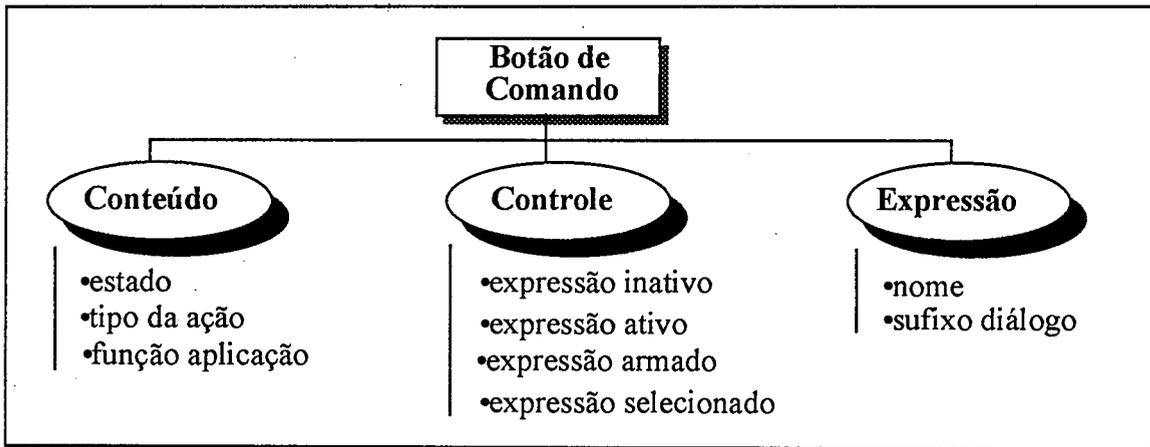


Fig 8.21 - A classe "Botão de comando"

8.3.2.2 - A opção de menu

A opção de menu corresponde a um campo uni-linear preenchido com o nome correto de um comando do sistema. Este é enviado para a interpretação pelo núcleo funcional assim que a opção seja selecionada pelo usuário. As opções de menu não ocorrem isoladas, pois são definidas como integrantes de um todo chamado menu.

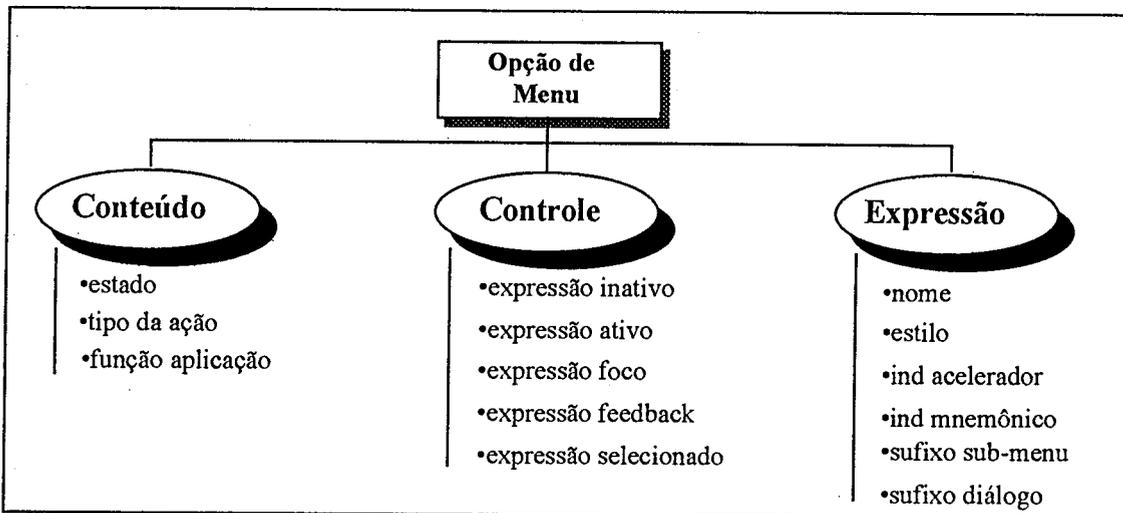


Fig 8.22 - A classe "Opção de Menu"

Elas podem ser classificadas como: opção de comando, opção de diálogo e opção de sub-menu. As opções do primeiro tipo acionam um comando da aplicação. As de diálogo acionam a apresentação de uma caixa de diálogo para precisar os parâmetros do comando. As de sub-menu causam a apresentação de um outro menu, o que permite ao usuário afinar a sua escolha em termos de opções de comandos. Estes diferentes tipos de opções devem ser

apresentados diferentemente. As opções que acionam caixas de diálogo devem ser seguidos de pontos de continuação "...". Aquelas associadas à sub-menus podem apresentar um marcador em forma de flecha, alinhado pela margem direita do menu.

As opções de menu podem assumir os estados "inativo", "ativo", em "foco", em "seleção" e "selecionado". Estas devem ser adequadamente representadas.

A seleção pode ser feita através do mouse ou do teclado. Na interação por mouse a área de seleção deve ser suficientemente ampla. A interação por teclado possibilita dois tipos de atalhos; os aceleradores e os mnemônicos. Um acelerador é um dispositivo que permite acionar uma opção de menu a partir de uma tecla de função ou de uma combinação de teclas. Isto ocorre independentemente da opção estar ou não visível. Neste caso a indicação da tecla associada deve fazer parte do rótulo da opção, geralmente à direita do nome da opção. Os mnemônicos acionam opções de menu que estejam visíveis. Trata-se de uma combinação de teclas envolvendo ou a letra inicial da opção ou uma outra que lhe seja marcante. Os mnemônicos devem estar representados pelo sublinhamento da letra definida no nome da opção. Seja qual for o meio de interação, somente a primeira letra do nome da opção deve estar em maiúsculo.

As opções podem assumir formas gráficas, tipográficas ou mistas. No caso de palavras, as denominações devem começar por um verbo, usual e simples, e devem evitar as abreviações. As opções de menu inativas devem estar diferenciadas coerentemente, isto é, de uma maneira dissimulada.

8.3.2.3 - O grupo de comandos

O grupo de botões de comando deve ser definido nas situações em que as opções de comandos possíveis forem em número reduzido. Seu arranjo pode seguir duas regras. Eles podem ser alinhados verticalmente e à direita do objeto a que fazem referência, ou horizontalmente e abaixo deste objeto. Um botão "par défaut" deve ser declarado e posicionado ao alto, se a orientação for vertical, ou à esquerda, no caso de uma orientação horizontal.

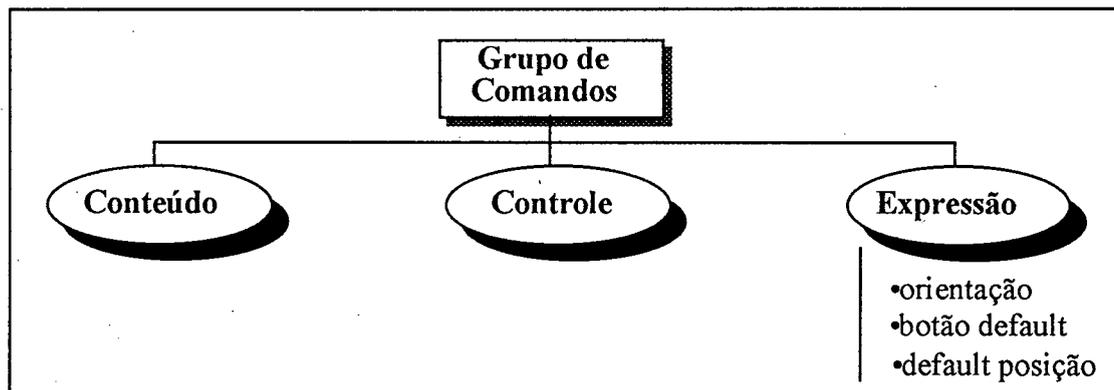


Fig 8.23 - A classe "Grupo de Comandos"

8.3.2.4 - O painel de menu

Um painel de menu corresponde a uma lista de seleção cujos itens são opções de menu ligadas a comandos, a caixas de diálogo e a outros menus. Em um menu as escolhas só podem ser simples. Todas as opções de um menu devem ser apresentadas simultaneamente, não sendo recomendados os dispositivos de navegação.

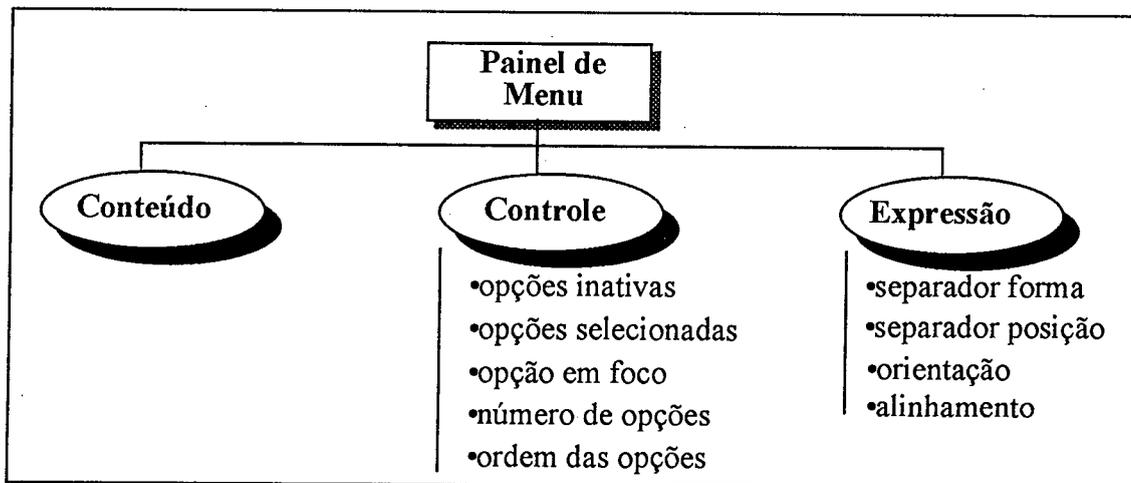


Fig 8.25 - A classe "Painel de Menu"

As opções devem refletir as necessidades da tarefa, devendo ser desprezadas as não pertinentes. Elas devem ser arranjadas segundo a ordem lógica da tarefa ou segundo a frequência de uso. Os menus muito numerosos devem ser organizados em grupos separados por um traço cheio ou uma linha em branco. As opções inativas não podem ser alvo de seleção. A orientação preferencial de um menu deve ser a vertical. A orientação horizontal deve ser considerada quando o número e o comprimento das opções for reduzido, como no caso da barra de menu.

Um painel de menu pode ser estático ou dinâmico. Os estáticos permanecem fixos em uma determinada posição da tela. Os dinâmicos só aparecem nela enquanto o usuário efetua uma seleção. Os menus dinâmicos, "de desdobrar" (pull-down), de "cascata" (cascading) e os de "sobre-apresentação" (pop-up) se diferenciam devido à forma de acionamento. O menu de desdobrar é apresentado na tela através do acionamento de uma opção de menu principal. O menu em cascata designa um menu de desdobrar que é apresentado a partir do acionamento de um outro menu de desdobrar. O surgimento na tela de um menu em sobre-apresentação (pop-up) independe de outra opção de menu. Ele ocorre simplesmente através do acionamento de um determinado botão do mouse. As opções de um menu pop-up variam conforme a posição do cursor na tela no momento em que ocorreu o acionamento. O painel de menu dinâmico deve ser apresentado próximo do local onde seu acionamento ocorreu. Os itens mais frequentes devem ocupar as primeiras linhas e sua orientação deve ser vertical com alinhamento pela esquerda.

Sob o ponto de vista da tarefa, os painéis podem ser principais e secundários. O menu principal representa o ponto de partida para a tarefa. Suas opções devem cobrir todas as alternativas em termos de ações básicas. Os menus secundários são acionados através de uma opção de menu e permitem ao usuário afinar a escolha de um comando.

Sob o ponto de vista do raciocínio por regras, os painéis de menus apresentam duas especializações; as páginas de menu e as barras de menu.

8.3.2.4.1 - A página de menu

A página de menu é característica dos chamados menus arborescentes. Trata-se de uma estrutura de encadeamento de sucessivas páginas de menus com objetivo típico de recuperação da informação em grandes bases de dados. Os aspectos principais destas estruturas de menus correspondem à largura (número de alternativas por página) e à profundidade (número de páginas por busca).

Uma página de menu deve possuir um título significativo, convite à interação (prompt), numeração e um histórico descritivo dos passos percorridos na interação.

O alinhamento do título e das opções de menus de página inteira deve ser definido pela esquerda. O Prompt deve ter a forma de um símbolo especial que lhe deve ser exclusivo e invariável. Sendo a seleção por teclado, as opções devem ser rotuladas com números e não

com letras. Esta numeração deve ser alinhada pela direita e separada do nome da opção através de um ou dois espaços. Ela deve partir sempre do número 1. No caso das letras serem definidas para a seleção por teclado, a numeração deve ser alinhada pela esquerda.

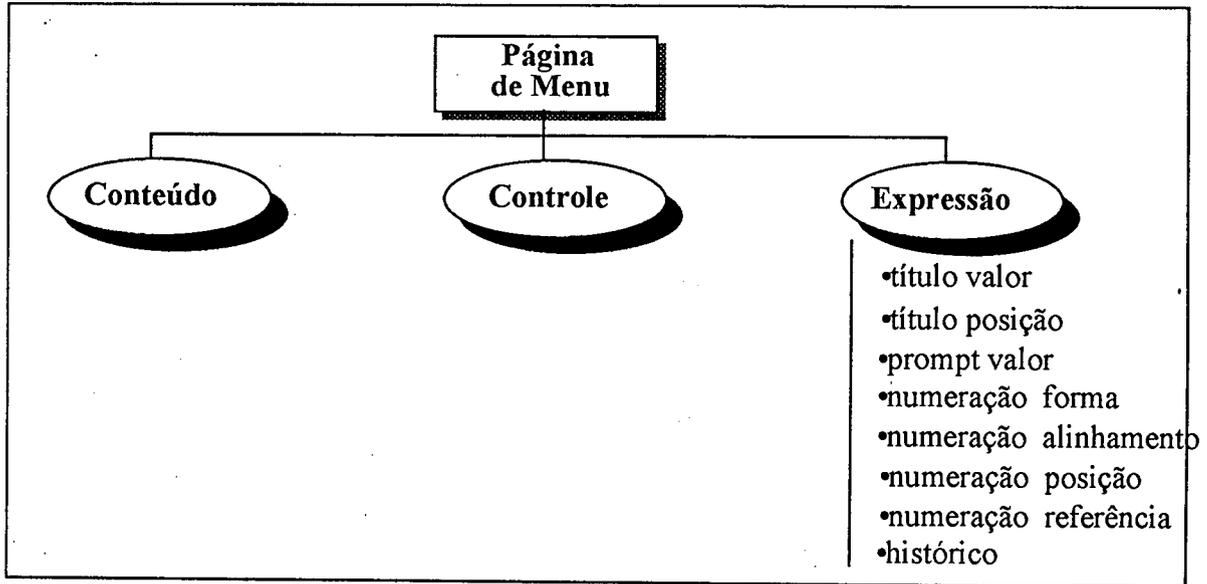


Fig 8.25 - A classe "Página de Menu"

8.3.2.4.2 - A barra de menu

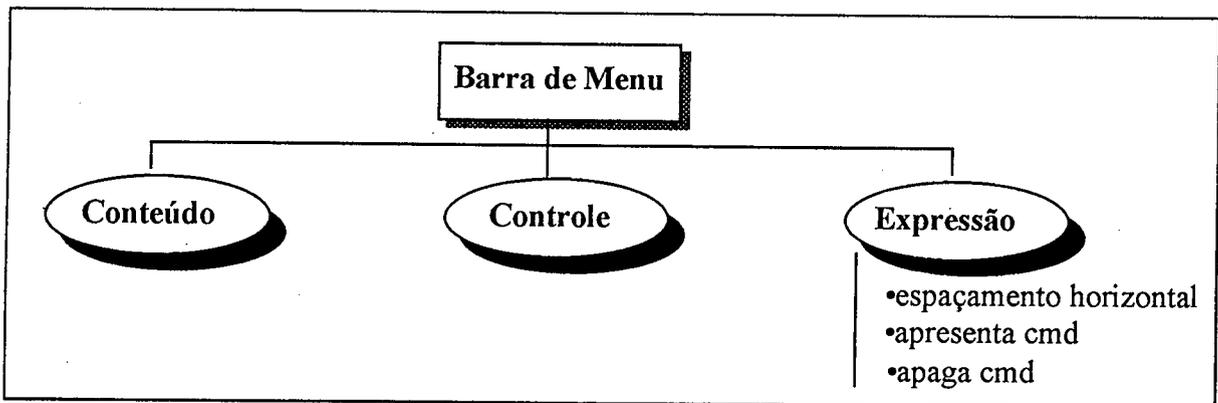


Fig 8.26 - A classe "Barra de Menu"

A barra menu tem orientação horizontal e deve ser posicionada no alto da tela ou da janela da aplicação. Ela apresenta os comandos de base de um sistema ou de um aplicativo que não devem ultrapassar as oito opções. Elas não comportam qualquer tipo de navegação e sua retirada da tela só se justifica para sistemas destinados ao grande público ou se expressamente comandada pelo usuário. Os nomes de suas opções devem ser palavras substantivas, não compostas e curtas, separadas por ao menos três espaços em branco. Se a ordem relativa à

tarefa ou à frequência de uso não podem ser definidas, um ordenamento alfabético deve ser empregado.

8.4 - Os Painéis de Controle

Os "Painéis de Controle" são OIA^e compostos que fornecem ao usuário um cenário adequado, em termos dos diferentes tipos de mostradores, de controles e de comandos necessários para a realização de sua ação ou tarefa.

Eles estão divididos em Janelas e Caixas de Diálogo. As janelas correspondem à expressão global de aplicativos e de documentos. As Caixas de diálogo fornecem informação e apoiam as tarefas e ações individuais.

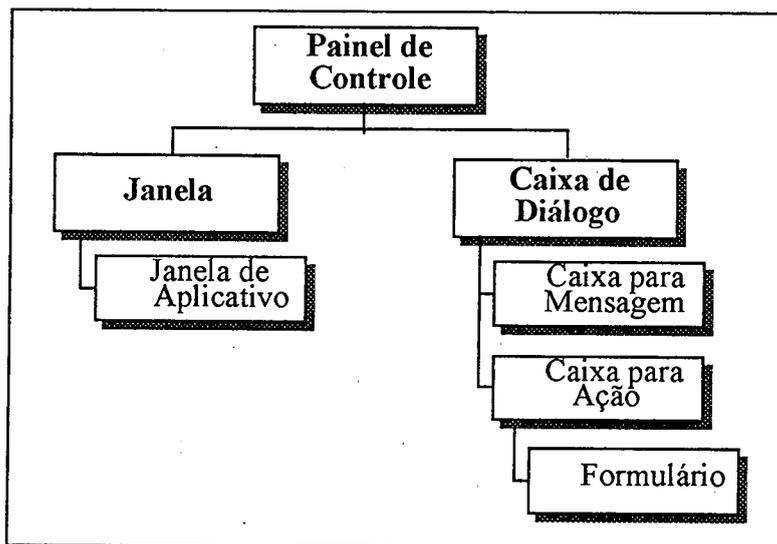


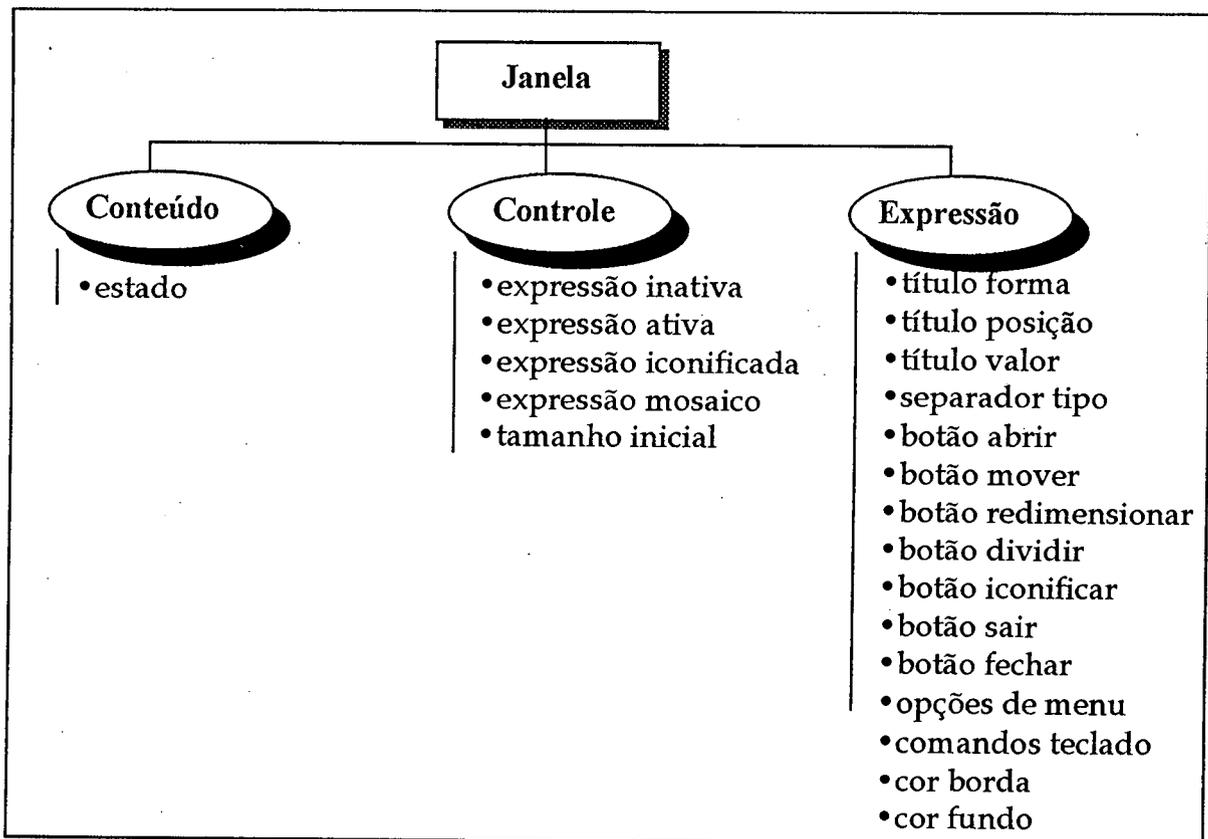
Fig 8.27 - As classes de Painéis de Controle

8.4.1 - A Janela

A janela gerada por um programa aplicativo pode assumir os estados "inativo", "ativo", "mosaico" e "iconificado". A cada um deles deve corresponder uma representação adequada. A janela ativa possui o foco das ações do usuário até que um outro foco seja definido. Ela é comumente colorida com cores densas. Os tons cinza são empregados para as janelas inativas. Na apresentação das janelas justapostas as barras de menu podem ser dispensadas.

Toda a janela deve possuir um título alfanumérico curto e significativo. Em caso de restrição de espaço, ele pode ser abreviado para que não ultrapasse os 4 caracteres. A posição do título deve ser mantida inalterada para todas as janelas do sistema. Estas devem propor

botões e/ou opções de menu e/ou comandos de teclado associados às ações de abertura, fechamento, de movimentação, de redimensionamento, de iconificação, de segmentação e de passagem para uma outra janela. As janelas de documentos devem apresentar barras de rolamento preferencialmente verticais. Nelas deve ser previsto os botões para a segmentação da janela. Esta separação deve ser indicada por uma linha cheia ou tracejada. A dimensão inicial proposta para uma janela deve ser compatível com as disponibilidades em termos de tela física, ficando à cargo do usuário o seu redimensionamento.



Fug 7.? - A classe Janela

8.4.1.1 - A janela de aplicativo.

A janela do aplicativo corresponde a uma área do terminal físico alocada para uma aplicação específica ou para o programa gerente de janelas (Finder, Presentation Manager, ...). Estes aplicativos podem por sua vez, criar diversas outras janelas de documentos que vão coexistir na tela. Em vista desta possibilidade, toda janela de aplicativo deve possuir uma opção de menu para o controle da disposição das janelas secundárias. Esta opção desdobrada deve incluir a lista dos documentos abertos e a lista de comandos relativos à disposição das janelas; justaposição, superposição, ou em cascata. A justaposição deve ser preferida quando as

manipulações sobre as janelas não forem frequentes ou quando o usuário for iniciante. Casos contrários, a opção preferível deve ser a superposição. A disposição em cascata pode tornar explícitas as interrelações de dependências entre janelas do mosaico. Aconselha-se estabelecer um limite para a quantidade de janelas abertas, que não deve ultrapassar as 7 janelas.

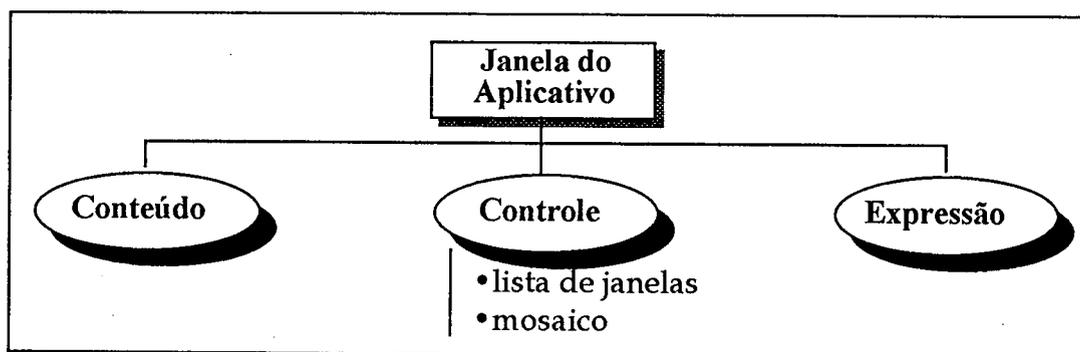


Fig 8.29 - A classe Janela do Aplicativo

8.4.2 - A caixa de diálogo

A caixa de diálogo corresponde a uma janela especialmente destinada à apresentação de mensagens e/ou de comandos para ações.

As caixas de diálogo podem ser modais ou não modais. Elas são modais quando exigem uma resposta do usuário, que fica impedido de qualquer outra ação até que isto aconteça. As não modais permitem que o usuário trabalhe sobre outros objetos de uma outra janela ou caixa de diálogo, enquanto que aguardam uma ação sua.

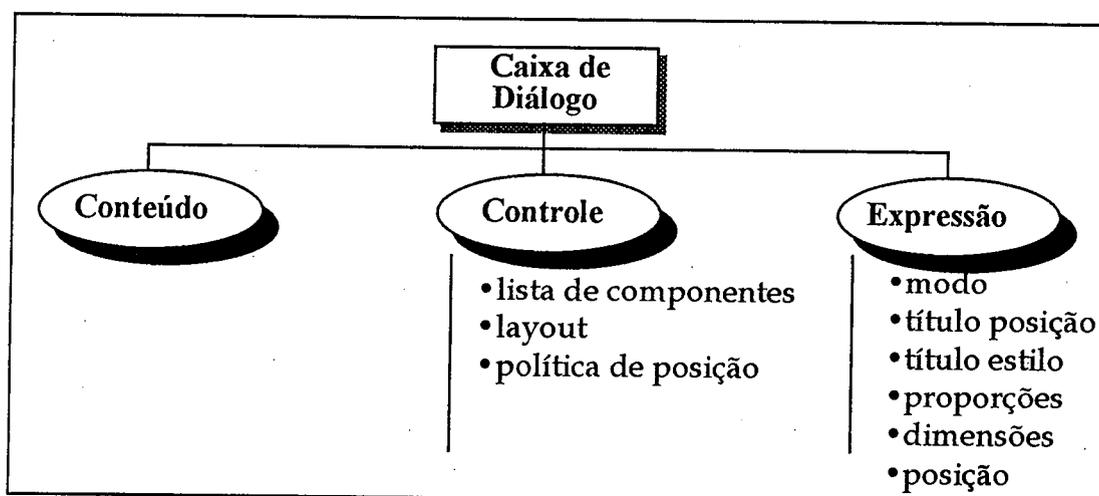


Fig 8.30 - A classe Caixa de Diálogo

As dimensões de uma caixa de diálogo devem ser definidas em função de seus componentes. A

relação entre o comprimento e altura deve se aproximar do chamado número de ouro; $1/1,618$. A política referente à sua aparição na tela pode ser de três tipos; padronizada, atraída pelo objeto a que se refere ou alinhada com o título da caixa de diálogo que lhe é superior. Seu título deve ser centrado na margem superior ou alinhado pela esquerda e a distribuição de seus integrantes pode seguir o desenho de uma grade. A densidade final de uma caixa de diálogo não pode exceder os 40 %.

8.4.2.1 - As caixas de mensagem

As caixas de mensagem fornecem informações e instruções ligadas à condução, à ajuda, às advertências, aos alarmes e aos erros na interação. Elas são do tipo modais, exigindo que o usuário tome seu conhecimento para permitir a sequência da interação. A mensagem deve ser adequada ao tipo de situação. Aconselha-se filtrar as mensagens provenientes do sistema operacional, do ambiente gráfico ou mesmo da aplicação e traduzi-las em termos da linguagem natural. O botão "ok" deve ser previsto em toda a caixa de mensagem como meio de receber a confirmação do usuário.

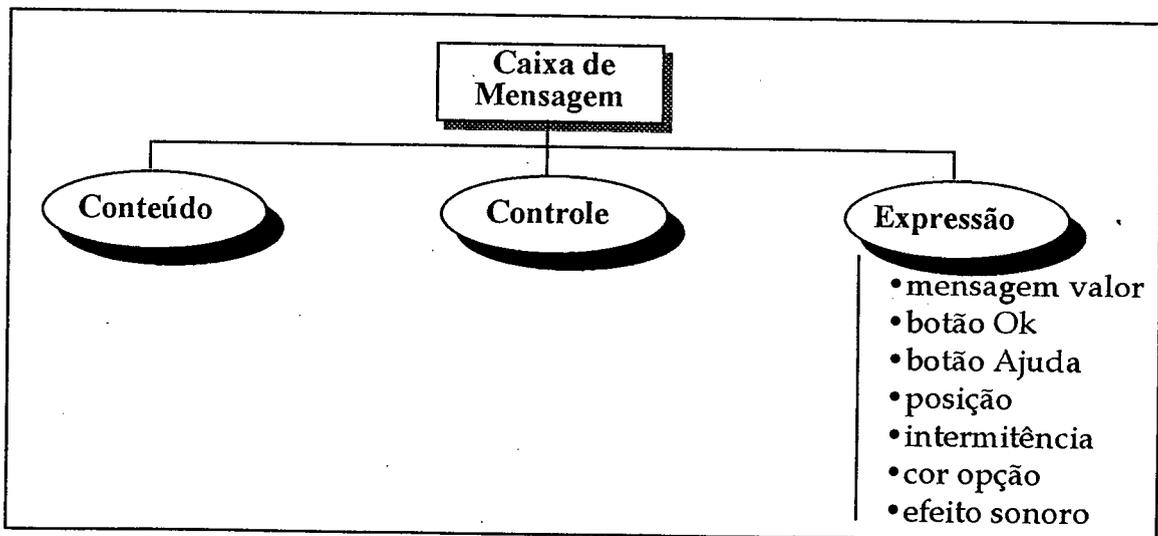


Fig 8.31 - A classe Caixa de Mensagem

Quando a situação for de alarme as caixas devem estar destacadas através de uma localização central da tela, do emprego da cor vermelho, da intermitência ou do acionamento de um efeito sonoro. Numa situação de advertência elas devem explorar o estereótipo do amarelo. As caixas de mensagens de erro devem propor sempre um botão "Ajuda".

8.4.2.2 - A caixa de Ação/Tarefa

A caixa de ação/tarefa proporciona os controles e comandos específicos para a introdução de parâmetros e para o acionamento da ação ou da tarefa que a define. Este tipo de caixa deve abrigar os botões de comando para validar, para aplicar imediatamente e para cancelar uma ação. Um deles deve ser definido "par default" e diferenciado apropriadamente. No caso de ações destrutivas, a opção "par default" deve recair sobre sua anulação e não sobre a própria ação. Também devem ser previstos botões de ajuda e de fechamento da caixa. A definição do layout destes botões deve minimizar os movimentos do mouse.

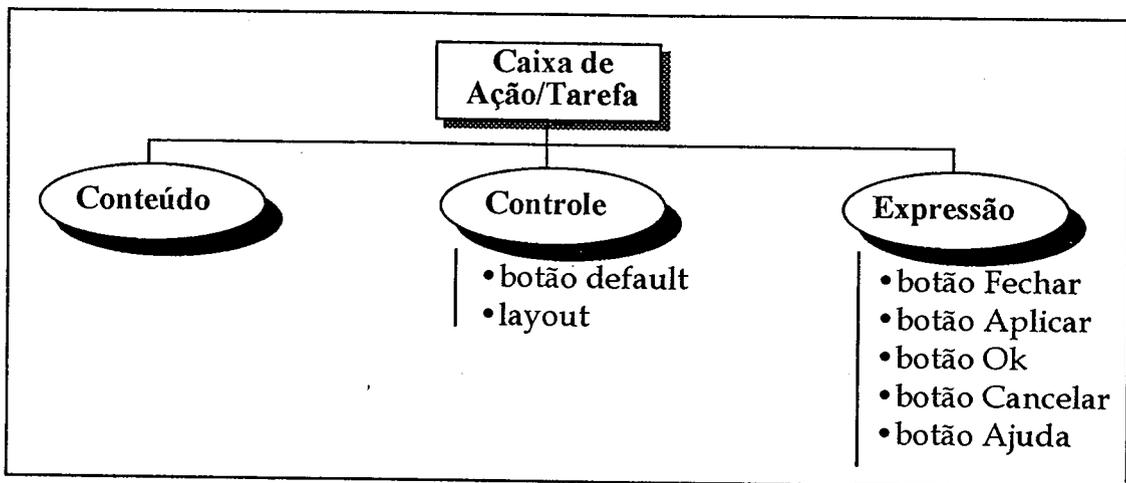


Fig 8.32 - A classe Caixa de Ação/Tarefa

8.4.2.3 - A caixa de Formulário

O formulário proporciona um painel para a apresentação e para a entrada de dados que apresenta um layout compatível com os documentos físicos manuseados pelo usuário em sua tarefa. A exemplo da classe de controles, uma caixa de formulário possui um mecanismo de proteção contra a modificação de dados considerados críticos para o sistema. Ao ser acionado este mecanismo torna o formulário insensível às ações do usuário.

O início das ações de entrada deve se dar a partir do campo localizado mais ao alto e à esquerda do painel. Esta posição inicial deve ser mantida de forma consistente durante a interação. Os campos devem estar adequadamente identificados, sendo que seus rótulos podem conter informação sobre as unidades e os valores aceitáveis para os dados a entrar. O alinhamento dos rótulos deve ser feito pela esquerda. Caso eles se apresentem comprimentos

muito diferentes seu alinhamento pode ser feito pela direita. O mecanismo de navegação entre os campos deve ser de simples operação. Aqueles contendo dados críticos para o sistema devem ser identificados e protegidos contra acidentes de operação. Um efeito sonoro pode ser empregado para informar sobre a proteção dos campos toda a vez em que o usuário tentar um acesso. Um formulário deve contar com a participação de mostradores de dados sobre o estado do sistema, linhas e até mesmo áreas de comandos. Estes devem estar localizados na parte inferior do painel. Esta mesma região deve ser reservada para a apresentação caixas de mensagens de erro. O registro dos dados entrados em um formulário não deve ser acionado como efeito colateral da ação de entrada de algum dado. O usuário deve explicitamente solicitar este registro através de um botão de comando. Este deve ter uma denominação coerente. Aconselha-se que seu rótulo contenha a palavra "Entre". Da mesma forma deve ser previsto um botão para cancelar este comando. Após o registro das entradas o sistema deve preencher com traços "----" os campos desconsiderados pelo usuário. O alinhamento dos dados numéricos deve ser feito a partir do ponto decimal.

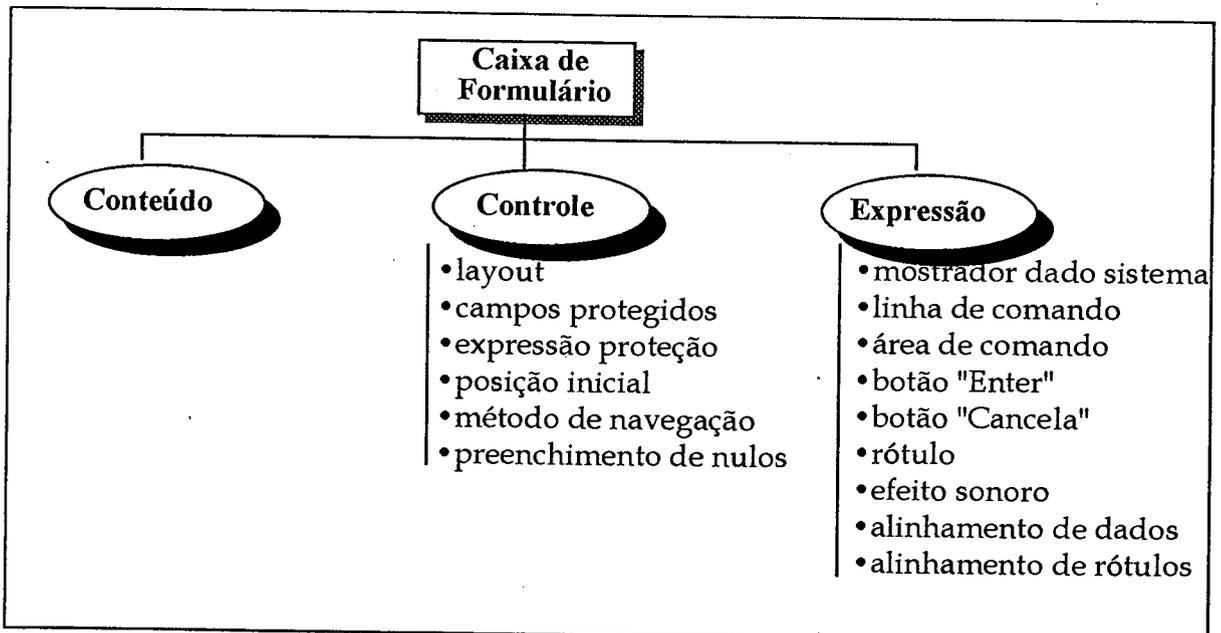


Fig 8.33 - A classe Caixa de Formulário

8.5 - O resumo do capítulo

✓ A definição de classes de funções baseia-se na interação dos OIA^e com o usuário e com a aplicação. Os exemplares destas classes modelam os componentes de uma ferramenta genérica; mostradores, controles, comandos e painéis de controle.

✓ Os mostradores são interadores que apresentam a capacidade de "acionar" o usuário através da apresentação das diferentes classes de dados e de informações de que ele necessita para pensar sua tarefa. Por outro lado eles são acionados por um programa aplicativo, que define quando, como e com quais valores os mostradores devem ser propostos ao usuário.

✓ Os controles são OIA^e sensíveis às ações do usuário, lhe proporcionando facilidades em termos de edição, seleção e manipulação direta. Esta classe de objetos não têm os recursos para acionar uma funcionalidade da aplicação. Assim, ela possibilita a entrada de dados que servirão como parâmetros para a ativação posterior de um comando.

✓ A classe de Comandos é uma especialização da classe de Controles cujos elementos abrigam chamadas para as ações da aplicação.

✓ Os "Painéis de Controle" são OIA^e compostos que fornecem ao usuário um cenário adequado, em termos dos diferentes tipos de mostradores, de controles e de comandos necessários para a realização de sua ação ou tarefa.

9 - DISCUSSÕES

Esta monografia se encerra com as discussões referentes as propostas apresentadas e as perspectivas em termos de pesquisas futuras ligadas aos objetos de interação abstratos ergonômicos.

9.1 - O resumo das propostas apresentadas.

A problemática geral deste trabalho refere-se à consideração efetiva dos preceitos da ergonomia cognitiva nas decisões de projeto de interfaces homem-computador. A forma de apoio pretendida refere-se a um ambiente computacional para a geração de interfaces ergonômicas, cuja base de regras é derivada das recomendações ergonômicas.

A identificação do conjunto de objetos de interface para o raciocínio ergonômico representa o objetivo específico deste trabalho. Face ao grande número de ambientes de programação e de objetos neles disponíveis, a primeira decisão se impõe. Quais objetos devem compor o ambiente de apoio pretendido? A resposta aponta para os objetos de interação portáteis ou abstratos, definidos, seja a partir do conjunto união ou intersecção dos objetos de interação dos diferentes ambientes.

9.1.1 - OIA^e como facilitador do raciocínio ergonômico

Verifica-se entretanto a impossibilidade de estabelecer uma correspondência efetiva entre as recomendações ergonômicas e um conjunto de objetos e de atributos abstratos. De fato, estas recomendações se referem a componentes de interface que existem nos modelos mentais dos ergonomistas e não em um ambiente de concepção. É necessário então definir um outro conjunto de objetos de interação, cujo nível de abstração se aproxime do verificado nas recomendações ergonômicas. Os OIA^e objetos de interação abstratos ergonômicos e seus atributos têm suas origens nas próprias recomendações. Eles são abstratos pois estão desvinculados de qualquer ambiente de concepção possível e são ergonômicos pois são derivados do exame de recomendações ergonômicas.

Apesar delas serem repletas de indicações sobre classes e atributos de componentes de interface, as recomendações ergonômicas nada indicam entretanto, sobre como organizar estes componentes, seja a nível de atributo seja a nível de classe. Estas definições estruturais são de

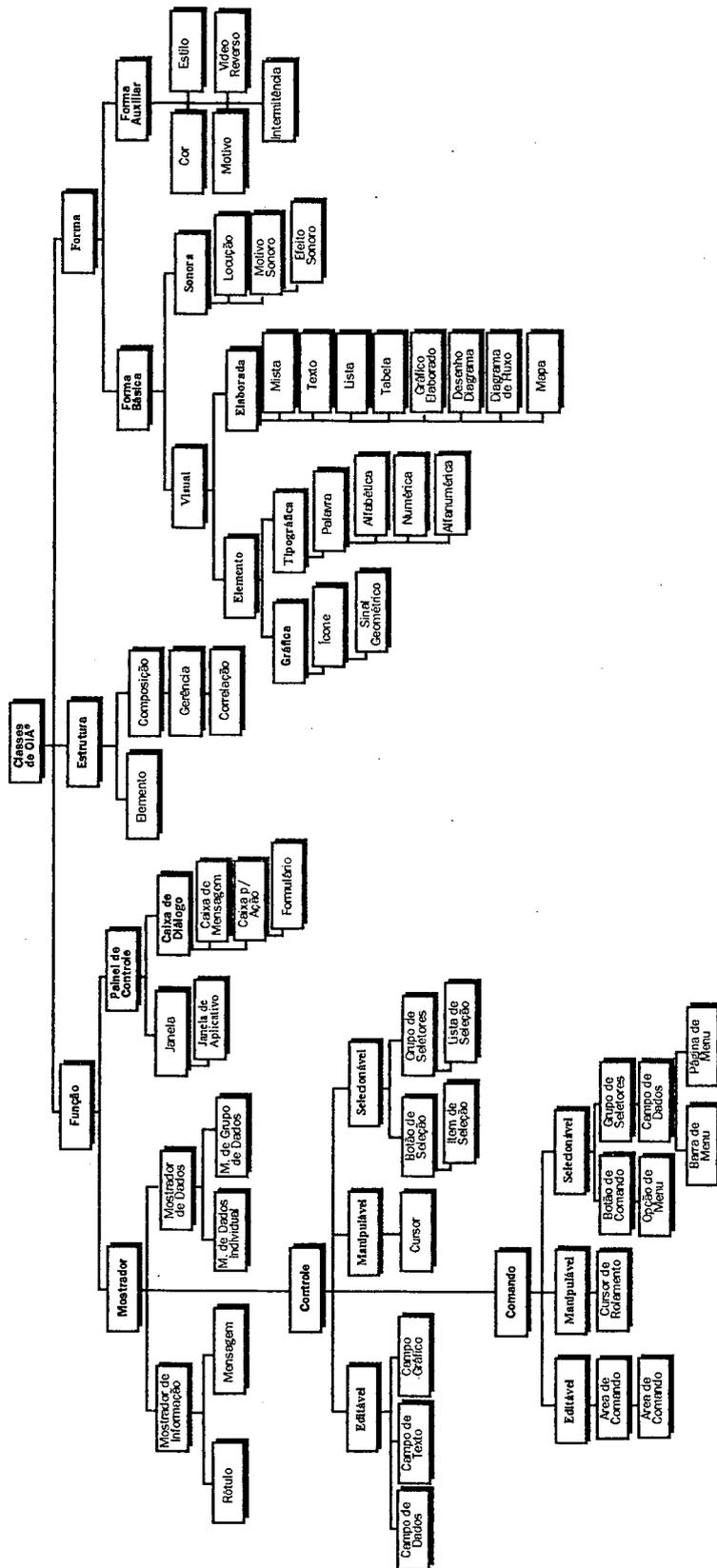


Fig 9.1 - As classes de objetos de interação abstratos ergonômicos - OIA^c -

fundamental importância para alcançar o êxito pretendido na identificação dos objetos de interface e de seus atributos ergonômicos.

Um OIA^e apresenta a morfologia tridimensional conteúdo-controle-expressão. Características em dimensões diferentes aumentam as possibilidades de correspondências com as recomendações ergonômicas. Se a maior parte destas recomendações refere-se aos aspectos da apresentação dos objetos, são em bom número também, aquelas que referem-se aos conteúdos a veicular, e à maneira pela qual realizar esta codificação. Um OIA^e atrai de uma maneira efetiva boa parte das recomendações relativas aos aspectos da sintaxe e da semântica de um sinal.

Outra característica explorada na definição das classes de OIA^e refere-se à repartição da semântica de um sistema interativo entre seus componentes, em especial entre os objetos interativos. Sob o ponto de vista de engenharia de software uma repartição de funções alivia o núcleo funcional e diminui a frequência de trocas entre componentes. Em consequência do esvaziamento do núcleo funcional ocorre o fortalecimento semântico dos OIA^e. Os conteúdos sugeridos nos textos das recomendações incluem o estado (ativo, inativo, em foco, em seleção, selecionado, default), a conformidade (ok, não usual, desvio, erro) e a implicação (alarme, anomalia, incidente). Verificar as conformidades, informar sobre as implicações e sobre as disponibilidades são ações típicas de elementos de interface. A identificação e a incorporação desta semântica pelos OIA^e favorece sua correspondência com as recomendações ergonômicas. Estes componentes encontram-se em um nível de abstração alguns degraus acima do verificado com os objetos de interação atuais.

Um OIA^e define um componente de interface cuja origem nas recomendações ergonômicas, cujas três dimensões de atributos e cujo elevado nível de abstração possibilitam uma correspondência efetiva entre suas características configuráveis e as recomendações ergonômicas. Sua aplicação em um ambiente lógico de apoio em ergonomia torna direto e descomplicado o raciocínio baseado em regras.

9.1.2 - OIA^e como recurso para a usabilidade

As classes e atributos de OIA^e apresentadas na segunda parte desta monografia representam os recursos de que dispõe o projetista para configurar a usabilidade de suas interfaces. Eles foram identificados no exame de cerca de 700 recomendações ergonômicas de

diversos autores e encontram-se distribuídos segundo classes de recursos funcionais, estruturais e formais.

Os recursos de formas básicas e auxiliares são definidos a partir de uma relação de determinação/subordinação. O mesmo ocorre entre estruturas elementares e compostas. Já as funções são definidas na metáfora dos componentes de uma ferramenta genérica informatizada: mostradores, controles, comandos e painéis de controle.

9.1.3 - OIA^e como sinal computacional

As soluções propostas para as questões de estrutura do modelo estão influenciadas pela perspectiva semiótica. Um sistema interativo é visto como um sistema de sinais, através dos quais o usuário aciona e interpreta os objetos de sua tarefa. Da combinação das contribuições da semiótica com técnicas de engenharia de software e das próprias recomendações ergonômicas, surge a proposta de OIA^e como sinal computacional. Um OIA^e representa a primitiva de uma possível programação orientada a sinais.

O paradigma de programação orientado a objetos se justifica na analogia com o mundo não informatizado, que é repleto de objetos. Ele representa uma ruptura na tradição, ao propor entidades englobando ao mesmo tempo a estrutura de dados e as ações possíveis sobre ela. Uma possível programação orientada a sinais se justifica na própria natureza de um sistema interativo, repleto de sinais.

A proposta da programação orientada a sinais é feita por Andersen (Andersen, 1990). Na analogia com a orientação à objetos...

"... cada sinal é caracterizado por atributos e por uma sequência de ações, muitas das quais o usuário pode ver ou escutar (os efeitos), mas algumas delas ele deve inferir. Os sinais organizam a substância de expressão gerada pela execução de um programa, e as transformações nesta substância são reflexo das ações dos sinais."

Este autor no entanto, não avança em sua proposta, que depende fundamentalmente da definição das primitivas de uma possível linguagem orientada a sinais...

"...Uma pré-condição para a construção de tal linguagem deveria estar no estabelecimento de um conjunto de taxemas através da análise dos sinais computacionais existentes."

A proposta dos OIA^e representa um avanço nesta direção, que é feito entretanto, à partir do exame das recomendações ergonômicas. Ela explora as noções de sinal e de agente PAC para compor uma entidade lógica com o objetivo básico de explicitar conteúdos, expressões e suas interações. Os sinais, assim como as recomendações ergonômicas estão ligados à mesma lógica; a utilização do sistema.

A um sinal, mais do que a um objeto, interessa os recursos básicos para ser destacado, discriminado e associado em um painel. Sua configuração pode ocorrer em dois planos, o do conteúdo e o da representação. Independentemente da aplicação, o projetista pode definir as expressões correspondendo aos diversos conteúdos. Pode-se mesmo pensar em componentes padronizados que tenham expressões destaque, associação e discriminação sugeridas "de fábrica". Isto garante uma uniformidade na expressão que vai contribuir para a consistência do projeto. Um segundo nível de configuração refere-se aos conteúdos que devem ser transmitidos aos usuários. Neste nível o projetista preocupa-se exclusivamente com as listas de objetos a destacar, a discriminar e a associar, despreocupando-se de como representar estas relações na tela.

9.2 - Pesquisas futuras

As frentes de trabalho abertas a partir da definição das classes de OIA^e incluem diversas atividades de pesquisa e desenvolvimento. As listadas no seguimento desta conclusão correspondem às atividades para as quais já se conta com a colaboração de pesquisadores bolsistas de mestrado do Curso de Pós-Graduação em Engenharia de Produção e Sistemas da Universidade de Santa Catarina.

9.2.1 - A validação do modelo de OIA^e

O tipo de validação já iniciada neste trabalho corresponde à validação por adesão. Ela confunde-se com o próprio desenvolvimento do modelo na medida em que consiste em verificar a capacidade dos OIA^e em atrair recomendações ergonômicas. A etapa de definições estruturais envolveu cerca de 200 recomendações utilizadas como controle para as diversas estruturas propostas. Tendo sido definidas a morfologia e a tipologia dos OIA^e realizou-se uma etapa de identificação de exemplares de classes, de sub-classes e de atributos. Foram então consideradas mais de 500 recomendações de diversos autores que aderiram, sem dificuldades à estrutura proposta. Foi possível comprovar então que esta estrutura é

suficientemente geral e aberta para definição de novas classes e de novos atributos de OIA^e. Este tipo de validação vai prosseguir com a atividade prevista de ampliação da base de recomendações que aderem ao modelo de OIA^e.

O segundo tipo de validação corresponde a validação pelo raciocínio ergonômico. Trata-se de uma atividade de manipulação envolvendo grupos de projetistas que serão convidados a avaliar, sob o ponto de vista ergonômico, a interface de um aplicativo. O projeto desta manipulação prevê: (i) a definição da população, (ii) a proposta um método de análise que uniformize a população, (iii) a definição dos limites da análise (iv) a definição das etapas de testes e (v) a definição dos métodos de análise dos dados. Espera-se comprovar nestes testes a validade do modelo de OIA^e como facilitador de qualquer tipo de raciocínio ergonômico.

9.2.2 - O desenvolvimento do aplicativo HyperAttribut

HyperAttribut é um aplicativo interativo cujo objetivo é de apoiar não só as atividades de desenvolvimento do modelo de OIA^e, como também aquelas de especificação e de avaliação de uma interface homem-computador.

Suas funções principais visam o registro, a classificação e a recuperação das recomendações ergonômicas segundo as classes e os atributos dos OIA^e já definidas ou ainda inéditas. HyperAttribut possibilita tanto a ampliação da base de recomendações que aderem ao modelo como a definição de novas classes de OIA^e dentro do modelo. Ela é constituída de duas pilhas HyperCard/HyperTalk denominadas "Recomendações" e "Classes".

A pilha de recomendações proporciona a interface com o expert em ergonomia que realiza o registro das recomendações. As cartas desta pilha respeitam a anatomia de um OIA^e, na medida em que propõe campos para receber além do nome da classe, os nomes dos atributos de conteúdo, de controle e de expressão a que uma recomendação se refere. As funcionalidades inseridas nestas cartas se destinam especialmente à ampliação do modelo e da base de recomendações a ele acopladas. As principais são as seguintes;

- a importação de recomendações a partir de um arquivo texto com a consequente criação de uma nova carta na pilha.
- o auxílio à decisão sobre qual classe de objetos e quais atributos desta classe são os mais adequados para receberem a recomendação atual. As funções são realizadas por

campos de rolamento que recuperam as classes de objetos e os diferentes atributos já definidos no modelo. Um histórico das decisões já realizadas a partir de recomendações semelhantes à que está sendo tratada completa este conjunto de funcionalidades.

- o registro, isolado ou em lotes, das recomendações ergonômicas, se necessário com a criação de novas classes e atributos na pilha "Classes". Esta constitui a principal função deste módulo, e a ela podem ser associadas a que modifica e a que anula completamente dos dados referentes a uma recomendação já registrada.
- diversos tipos de navegação com funções de; busca a partir de uma expressão qualquer; acesso a todas as cartas definidas para uma mesma classe de objetos na pilha "Recomendações"; acesso a uma determinada classe, ao "Índice" e ao "Sumário" da pilha "Classes".

As cartas da pilha "Recomendações" abrigam também campos para a edição de informações adicionais sobre as recomendações, como a referência, os comentários, os exemplos e contra-exemplos de sua aplicação, etc. Estes campos representam o apoio ao entendimento das recomendações.

A pilha "Classes" representa o módulo de recuperação das recomendações ergonômicas. Esta pilha possui cartas de três tipos. As cartas "Classe" com as classes de objetos, uma carta "Índice" e uma carta "Sumário". Nas cartas "Classe" estão registrados os três tipos de atributos de um OIA^e com as respectivas recomendações. Suas funcionalidades permitem;

- o exame dos atributos da classe através de sucessivos menus pull-down. Cada sequência de examinação é acompanhada da apresentação das recomendações que dizem respeito a cada opção de menu. No caso de objetos compostos o exame de seus atributos pode se estender a várias cartas dos componentes deste objeto. Neste caso, funções especiais de navegação são ativadas para que o examinador possa a qualquer instante retornar à classe do objeto composto inicial. No caso de sub-classes os atributos e recomendações herdadas de outras classes são igualmente apresentadas.
- o acesso às cartas que abrigam as recomendações mostradas na janela de informação que acompanha o exame dos atributos de uma classe. Esta funcionalidade tem o

objetivo de proporcionar ao examinador uma série de informações adicionais sobre a recomendação, como fonte, comentários, exemplos e contra-exemplos de sua aplicação; etc.

- a montagem de relatórios que, com diferentes níveis de detalhe, apresentam atributos, instâncias de atributos e recomendações já registradas em uma determinada classe.

Esta pilha propõe também funcionalidades voltadas para o desenvolvimento e para a manutenção do modelo e da base de recomendações. Elas são as seguintes;

- a montagem das heranças de novas classes em função das já existentes no modelo.
- a realização de modificações dos nomes dos objetos e dos atributos com a necessária repercussão sobre todo o sistema aplicativo.

Este aplicativo constitui a maquete de uma ferramenta de ajuda à especificação e à avaliação das interfaces com o usuário baseada no modelo de OIA^e. Ele oferece o acesso à informação ergonômica e um tipo de condução por objetos para as ações de especificação e de avaliação das interfaces homem-computador.

Pretende-se o desenvolvimento de uma nova versão de HyperAttribut em plataformas diversas (Macintosh, PC e estações de trabalho). A definição desta nova versão resultará da análise do desempenho da maquete existente nos testes de validação do modelo de OIA^e pelo raciocínio ergonômico. A idéia é de aproveitar a mesma manipulação para verificar a facilidade de uso e a adaptação à tarefa (utilisabilidade) da versão atual de HyperAttribut. Os dados desta manipulação definirão as características de sua nova versão, que será distribuída em shareware.

9.2.3 - A ampliação e o refinamento da base de recomendações ergonômicas

A pilha de recomendações ergonômicas registradas em HyperAttribut se aproxima atualmente das 700 cartas. Pretende-se a realização de uma etapa de trabalhos visando (i) a ampliação deste número, (ii) a tradução dos textos das recomendações para a língua portuguesa e (iii) o refinamento desta base pelo esclarecimento sobre as recomendações conflitantes.

9.2.4 - A ligação utilização/funcionamento

O confronto entre as lógicas de utilização e de funcionamento das interfaces homem-computador é o objeto de exame de uma atividade de pesquisa futura. Esta prevê a implementação de recursos de OIA^e e de suas correspondências com os objetos concretos.

A partir das classes de componentes funcionais pretende-se a construção de recursos de OIA^e e a montagem de uma "toolbox abstrata". Sua utilização por um grupo de projetistas deverá ser o objeto de um terceiro tipo de validação do modelo; a validação pelo uso. Ela se baseia na possibilidade de especificar uma interface completa através das classes de OIA^e identificadas.

Isto realizado, o próximo passo deve examinar as correspondências entre os OIA^e da toolbox abstrata com os objetos concretos de diferentes toolboxes. Uma estratégia a ser examinada consiste na utilização dos objetos de interação abstratos como elo de ligação entre os OIA^e e os objetos concretos.

9.2.5 - Em busca de uma nova dimensão de OIA^e ; os sequenciais

Os aspectos sequenciais de uma interface homem-computador podem ser configurados através dos chamados esqueletos de aplicação. Eles definem segmentos de diálogo que são propostos "prets à employer" aos projetistas de interfaces. Pode-se aplicar à tarefa de identificação dos objetos sequenciais e de seus atributos ergonômicos a mesma estratégia de ação seguida para os objetos estáticos. A hipótese é de se poder definir diálogo como um sinal, de morfologia tridimensional e cujas classes são definidas a partir das perspectivas funcional, estrutural e formal.

10 - Referências Bibliográficas

- Andersen, P. B. (1990). A Theory of Computer Semiotics: semiotic approaches to construction and assessment of computer systems. (First ed.). Cambridge, U.K.: Press Syndicate of the University of Cambridge.
- Andersen, P. B. (1990b). Towards an aesthetics of hypertext systems: a semiotic approach. (Venus Report No. 3). Departement of Information and Media Science - University of Aarhus.
- Andersen, P. B. (1993). The force dynamics of interctive systems: Towards a computer semiotics. pp. 36. Department of Information and Media Science - Universtity of Aarhus.
- Aschehoug, F., & Scapin, D. L. (1990). Base de recommandations ergonomiques pour l'évaluation et la conception d'interfaces utilisateurs (Document Interne No. INRIA - Institut National de Recherche en Informatique et en Automatique).
- Baar, D. d., Foley, J. D., & Mullet, K. E. (1992). Coupling application design and user interface design. In INTERCHI'92 - Human Factors in Computing Systems, 1 (pp. 259-266). Monterey, CA: ACM Press.
- Barthet, M.-F. (1988). Logiciels interactifs et ergonomie: modèles et méthodes de conception (first ed.). Paris: Bordas.
- Bass, L., & Coutaz, J. (1991). Developing software for the user interface (first ed.). Reading, Massachusetts: Addison-Wesley Publishing Company.
- Bastien, C. & Scapin, D. (1993). Human factors criteria, principles, and recommandations for HCI: methodological and standatdisation issues. (Internal Repport). INRIA
- Bastien, C. J. (1991). Validation des critères ergonomiques pour l'évaluation d'interfaces utilisateurs. (Rapport de Recherche No. 1427). INRIA.
- Blattner, M. M., Sumikawa, D. A., & Greenberg, R. M. (1989). Earcons and Icons: Their Structure and Common Design Principles. Human-Computer Interaction, 4(1), 11-44.

- Bodart, F., & Vanderdonckt, J. (1993a). Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection. In A. Wesley (Ed.), INTERCHI'93 - Human Factors in Computing Systems, 1 (pp. 424 - 429). Amsterdam: ACM Press.
- Bodart, F., & Vanderdonckt, J. (1993b). Guide Ergonomique de la présentation des applications hautement interactives. (1° ed.). Namur, Belgique: Presses Universitaires de Namur.
- Brown, C. M. (1988). Human-computer interface design guidelines (first ed.). Norwood, New Jersey: Ablex Publishing Corporation.
- Card, S., Moran, T. & Newell, A. (1983). The psychology of human-computer interaction (first ed.). Hillsdale: Lawrence Erlbaum Associates.
- Chapanis, A. & B., W. (1990). Specifying human-computer interface requirements. Behaviour & Information Technology, 9(6), 479-492.
- Chauvet, J.-M., Harman, D., & Jouhier, B. (1991). Open Interface: un outil de construction d'interfaces graphiques portables. Genie Logiciel & Systèmes Experts, 24, 90-102.
- Chevalier, J. (1980). Dictionnaire des symboles (first ed.). Paris: Robert Laffont.
- Coutaz, J. (1990). Interfaces homme-ordinateur: conception et réalisation. (first ed.). Paris: Bordas.
- Cybis, W. (1990). Abordagem ergonômica para a concepção/avaliação de um sistema de informação. In Congresso Internacional da Tecnologia do Software, Telemática e Informação, (pp. 1-16). São Paulo:
- Fairchild, K., Meredith, G., & Wexelblat, A. (1989). A Formal Structure for Automatic Icons. Interacting with Computers, 1(2), 131-140.
- Faveaux, L., & Yedid, C. (1991). Réflexion sur la mise en oeuvre d'une démarche ergonomique de conception d'interface. In Y. Quéinnec & F. Daniellou (Ed.), Congress of International Ergonomics Association, 1 (pp. 649-651). Paris: Taylor & Francis.
- Foley, J. D. & V. D., A. (1984). Fundamentals of interactive computer graphics (first ed.). Massachusetts: Addison-Wesley Publishing Company.

- Gould, J. D., Boies, S. J., & Lewis, C. (1991). Making usable, useful, productivity-enhancing computer applications. Communications of the ACM, 34(1), 74-85.
- Gray, P. D., Waite, K. W., & Draper, S. W. (1990). Do-It-Yourself Iconic Displays: Reconfigurable Iconic Representations of Application Objects. In Proceedings of IFIP INTERACT'90: Human-Computer Interaction (pp. 639-644).
- Green, A. J. K., & Barnard, P. J. (1990). Iconic Interfacing: The Role of ICON Distinctiveness and Fixed or Variable Screen Locations. In Proceedings of IFIP INTERACT'90: Human-Computer Interaction (pp. 457-462).
- Guiraud, P. (1983). La sémiologie (4° ed.). Paris: Presses Universitaires de France.
- Hollnagel, E. (1989). Information and reasoning in intelligent decisions support systems. In E. Hollnagel Mancini, G. & Woods, D. (Eds.), Cognitive engineering in complex dynamic London: Academic Press.
- ISO 9241 Part 10 (1993). Ergonomic requirements for office work with visual display terminals, Part 10 Dialogue Principles; Committee Draft ISO 9241-10
- ISO 9241 Part 14 (1993). Ergonomic requirements for office work with visual display terminals, Part 14 Menu Dialogues; Draft International Standard ISO 9241-14
- IBM (1989). Systems Application Architecture, Common User Access: Advanced Interface Design Guide. In Boca Raton: International Business Machines Corp.
- Johnson, J. (1992). Selectors, going beyond user-interface widgets. In INTERCHI'92 - Human Factors in Computing Systems, 1 (pp. 273-279). Monterey, CA: ACM Press.
- Johnson, J., Roberts, T., Verplank, W., Smith, D., Irby, C., Beard, M., & Mackey, K. (1989). The Xerox Star: a retrospective. IEEE Computer, 22(9), 11-29.
- Laville, A. (1977). Ergonomia (First ed.). São Paulo: Editora Pedagógica e Universitária Ltda.
- Loshe, G., Walker, N., Biolsi, K., & Rueter, H. (1991). Classifying Graphical INFORMATION. Behaviour and INFORMATION Technology, 10(5), 419-436.
- Morris, C. (1974). Fondements de la théorie des signes. Langages, 8(35), 15-21.

- Nadin, M. (1988). Interface design and evaluation - Semiotic implications. In H. R. Hartson & D. Hix (Eds.), Advances in human-computer interaction (pp. 376). Norwood, New Jersey: Ablex Publishing corporation.
- Norman, D. A. (1984). Cognitive engineering principles in the design of human-computer interfaces. In G. Salvendy (Eds.), Human Computer Interaction Amsterdam: Elsevier Science Publisher.
- Norman, D. A. (1991). Cognitive Artifacts. In J. M. Carrol (Eds.), Designing Interaction, psychology at the human-computer interface (pp. 17-38). Cambridge: Cambridge University Press.
- OSF (1990). Motif Style Guide (1.1 ed.). Cambridge, MA: Open Software Foundation.
- Microsoft Corporation(1987). Microsoft Windows: Software development kit: Application Style Guide (version 2).
- Pollier, A. (1991). Evaluation d'une interface par des ergonomes: diagnostiques et stratégies (Rapport de Recherche No. 1391). INRIA.
- Prieto., L. J. (1972). Messages et Signaux (first ed.). Paris: Presses Universitaires de France.
- Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs and symbols, and other distinctions in human performance models. IEEE Transactions on Systems, Man, and Cybernetics, SMC 13(3), 257-266.
- Ravden, S., & Johnson, G. (1989). Evaluating usability of human-computer interfaces (first ed.). Chichester: Ellis Horwood Limited.
- Richard, J.-F. (1983). Logique du fonctionnement et logique d'utilisation (Rapport de Recherche No. 202). INRIA.
- Richard, J.-F., Bonnet, C., & Ghiglione, R. (1990). Traité de Psychologie Cognitive : perception, action, langage(première ed.). Paris: Bordas.
- Rivlin, C., Lewis, R., & Davies-Cooper, R. (1990). Guidelines for Screen Design (first ed.). Cambridge: Blackwell scientif Publications.

- Rogers, Y. (1989). Icons at the Interface: Their Usefulness. Interacting with Computers, 1(1), 105-117.
- Sacre, B., Sacre-Provot, I., & Vanderdonckt, J. (1992). Une description orientée objet des objets interactifs abstraits utilisés en interfaces homme-machine. (rapport IHM/Ergo No. 10). Facultés Universitaires Notre-Dame de La Paix - Institut d'Informatique - Projet Trident.
- Scapin, D. L. (1986). Guide ergonomique de conception des interfaces homme-machine. (Rapport de Recherche No. 77). INRIA - Rocquencourt - France.
- Scapin, D.L. et al. (1988a) - La concéption ergonomique d'interfaces: problèmes de méthodes, Rapports de Recherche , Rocquencourt, France: INRIA.
- Scapin, D.L. (1988b) - Vers des outils formels de description des tâches, orienté conception d'interfaces, Rapports de Recherche , Rocquencourt, France: INRIA.
- Scapin, D.L. (1989a) - Guidelines for user-interface design: Knowledge collection and Organisation, Technical Raport TR.D12.1 Version Date 30/12/1989, Rocquencourt, France: INRIA.
- Scapin, D. L. (1989b) - MAD: Une méthode analytique de description des tâches. IN: Colloque sur l'engeniérie des interfaces homme-machine, Sophia-Antipolis, France, INRIA
- Scapin, D. L. (1990a) - Aiding mechanisms for the design of user interface: Some researche issues. IN: Proceedings of the First International Conference on Automation Technologie. p 587-593, Taiwan.
- Scapin, D.L. (1990b) - Decyphering human-factors recommendations, IN: Karwoski, W. & Rahimi, M. (Eds). Ergonomics of hybrid automated Systems II, Amsterdam: Elsevier Science Publisher.
- Scapin, D.L. (1990c) - Des critères ergonomiques pour l'évaluation et la concéption d'interfaces. IN: Actes du Congres de la SELF, Montréal.
- Scapin, D. L. (1990d). Organizing human factors knowledge for the evaluation and design of interfaces. International Journal of Human-Computer Interaction, 2(3), 203-229.

- Sebillotte, S. (1991). Decrire des tâches selon les objectifs des operateurs: de l'interview à la formalisation (Rapport de Recherche No. 125). INRIA.
- Shneiderman, B. (1987). Designing the User Interface: strategies for effective human-computer interaction (first ed.). Addison-Wesley Publishing Company.
- Smith, S. (1986). Standards versus guidelines for designing user interface software. Behaviour and Information Technology, 5(1), 47-61.
- Smith, S. L., & Mosier, J. N. (1986). Guidelines for designing user interface software No. ESD-TR-86-278). The MITRE corporation.
- Sousa, F. d., & Bevan, N. (1990). The use of guidelines in menu interface design: evaluation fo a draft standard. In D. Diaper (Ed.), Human Computer Interaction - INTERACT 90, (pp. 435-440). Elsevier Science Publishers.
- Valentin, A., & Lucongong, R. (1987). L'ergonomie des logiciels (first ed.). Paris: ANACT.
- Ware, C. (1992). The foundations of experimental semiotics: a theory of sensory and conventional representation. Experimental Semiotics(6), 1-13.
- Williges, B., & Williges, R. (1984). Dialogue design considerations for interactive computer systems. In F. A. Muckler (Eds.), Human Factors Review Santa Monica: The Human Factors Society, Inc.
- Winograd, T., & Flores, F. (1986). Understanding computers and cognition: a new foundation for design (first ed.). Norwwod, New Jersey: Ablex Publishing Corporation.
- Zanden, B. V., & Myers, B. A. (1990). Automatic, look-and-feel independent dialog creation for graphical user interfaces. In (pp. 27-34). ACM Press.

acionamento de uma expressão específica e predefinida pelo projetista. Sugere-se o emprego da cor amarela na definição da "expressão atualização".

Através da definição destas duas classes é possível exemplificar o efeito de repartição da semântica entre os componentes de um sistema interativo, que é proporcionado pela arquitetura PAC (Coutaz, 1990). Ele se verifica na delegação de uma parte das competências do núcleo funcional aos objetos de interação, com a vantagem de diminuir a frequência de trocas entre eles. Estes dois tipos de campos reivindicam para si a tarefa de inspecionar e de conduzir. Eles definem um nível de competência usualmente superior à dos objetos de interação. Este efeito de repartição semântica dá margem à formação do que se pode chamar de semântica da interação. Ela resulta da reunião de tarefas que deixam o núcleo funcional, sendo atraídas pelos OIA^e.

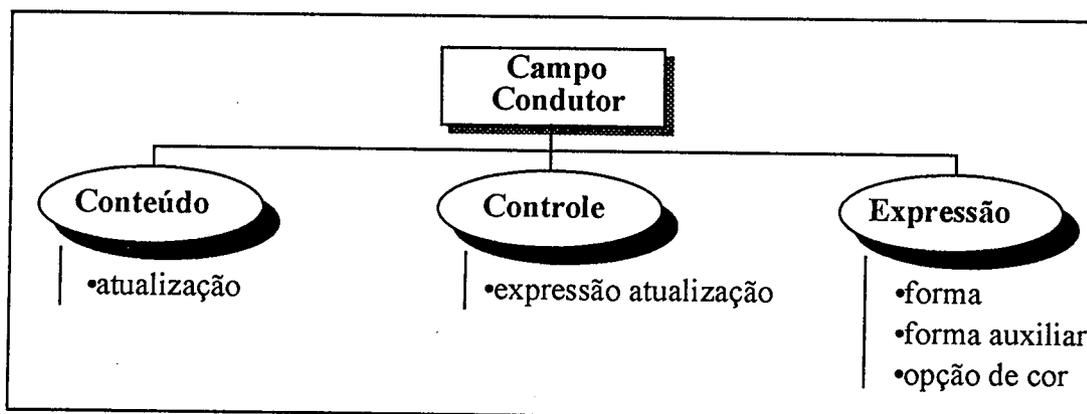


Fig 8.13 - A classe "Campo Condutor"

8.2.2.2 - O Campo de Texto

O campo de texto apresenta os recursos de edição multi-linha para receber do usuário dados na forma textual. Seu tamanho em termos do número e do comprimento de linhas deve ser adequado para proporcionar um desempenho eficiente na tarefa de entrada de textos. Para a facilidade de leitura o comprimento das linhas não deve exceder os 40 caracteres. Sob o ponto de vista de edição, o projetista pode especificar um cursor intermitente para auxiliar na localização do ponto de inserção de texto.

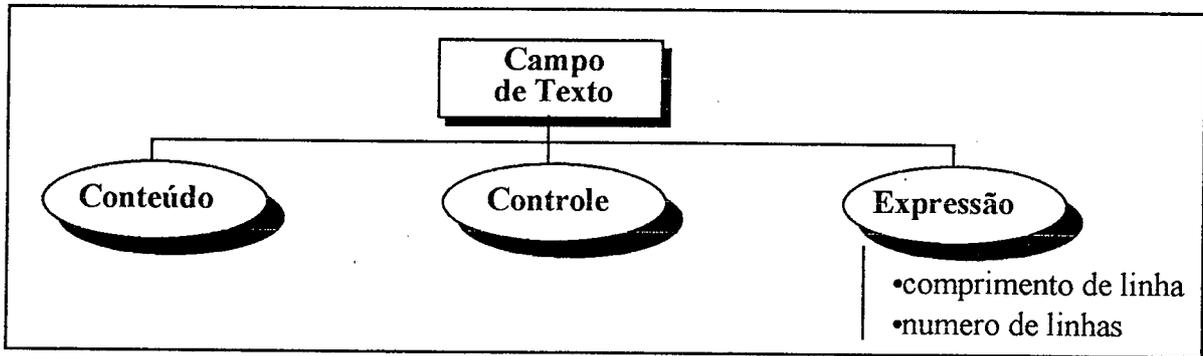


Fig 8.14 - A classe "Campo de Texto"

8.2.2.4 - O Campo Gráfico

O campo gráfico se define nos recursos de edição gráfica que proporciona ao usuário. As recomendações ergonômicas são econômicas em relação à esta classe de objetos. Eles se referem às diferentes formas do cursor, como "+" ou "†", face aos diferentes tipos de tarefa. O centro de rotação de um objeto deve coincidir com o centro do próprio objeto e uma função de zoom deve ser prevista.

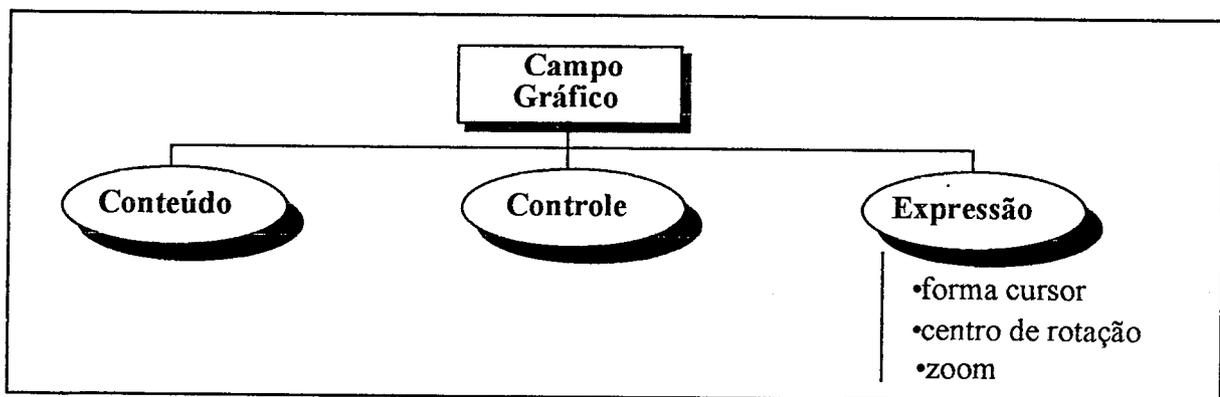


Fig 8.15 - A classe "Campo Gráfico"

8.2.3 - Os controles selecionáveis

Os seletores ou campos de seleção, permitem a entrada de dados através da indicação dos valores predefinidos que neles residem.

8.2.3.1 - O botão de seleção

O botão de seleção corresponde a um campo de seleção que pode assumir três estados: um estado "ativo", onde a seleção é possível; um estado "inativo", onde a seleção não é

possível e finalmente o estado "definido", onde a seleção já ocorreu. Estes devem ser representados através de formas coerentes e suficientemente distintas.

Um botão de seleção apresenta o seguinte comportamento imediato. No estado "inativo" ele não pode ser selecionado. Se selecionado a partir de seu estado "ativo", ele assume o estado "definido". Se selecionado a partir do estado "definido" ele volta ao estado normal "ativo". O comportamento em um grupo define os dois tipos de botões de seleção; o não exclusivo (toggle) e o exclusivo (radio). Em um grupo de botões exclusivos, só um deles pode estar definido ao mesmo tempo. A seleção de um outro botão causa o "desligamento" do anterior. Em listas de botões não exclusivos, diversas opções podem estar definidas simultaneamente. Um botão deste tipo pode ser selecionado livremente sem consequências sobre os outros componentes do grupo.

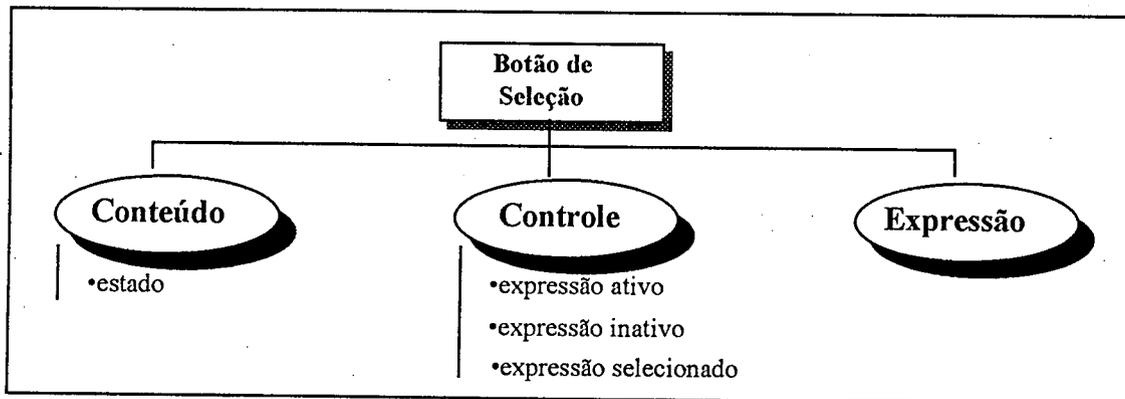


Fig 8.16 - A classe "Botão de Seleção"

8.2.3.2 - O item de seleção

O item de seleção corresponde a um campo para a seleção de um dado de forma elementar gráfico, elementar tipográfico ou misto. O item de seleção invariavelmente integra uma lista de seleção.

Ele pode assumir os estados "inativo", "ativo", em "foco" e "selecionado", para os quais deve ser previstas formas de expressão coerentes e diferenciáveis. No estado ativo o item se encontra sensível a uma ação de seleção. No estado inativo a seleção não é possível. O estado "em foco" corresponde ao momento anterior a uma seleção, quando o usuário posiciona o cursor sobre o item e prepara-se para selecioná-lo. A expressão de "feedback" de seleção informa que uma seleção acaba de ocorrer. A expressão "selecionado" indica ao usuário que aquele item já se encontra definido.

O vídeo reverso pode ser empregado como expressão "feedback". Marcadores, com a forma de asteriscos, flechas ou símbolos especiais podem ser utilizados para as expressões "selecionado". A expressão "inativo" é tipicamente representada pelo tom acinzentado de suas formas.

Um item de seleção apresenta o comportamento básico de um botão, podendo ser exclusivo (radio) ou não exclusivo (toggle).

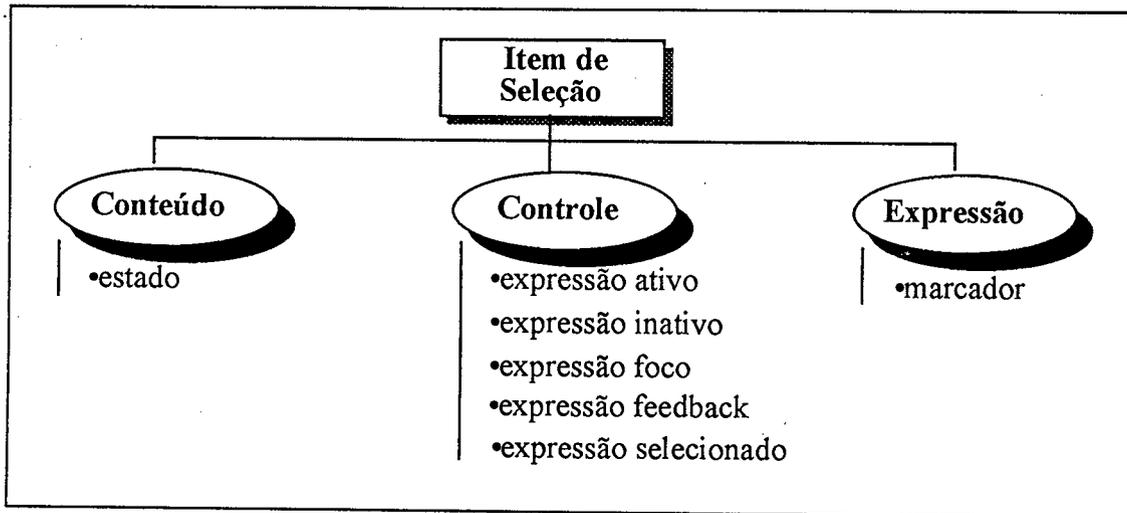


Fig 8.17 - A classe "Item de Seleção"

8.2.3.3 - O grupo de Seletores

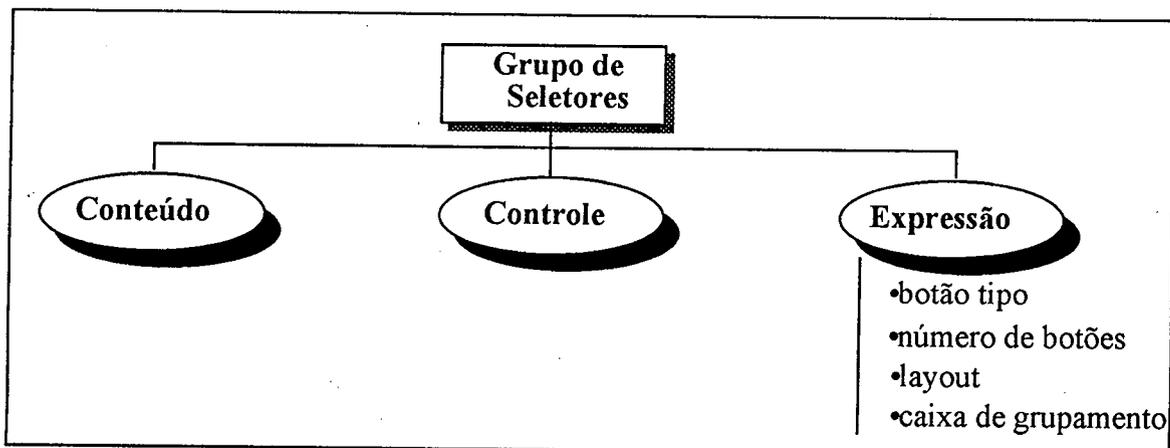


Fig 8.18 - A classe "Grupo de Seletores"

O grupo de botões reúne um máximo de sete botões de seleção ou de rádio que devem estar distribuídos equidistantemente. A caixa de grupamento só deve ser utilizada se eles foram mais do que quatro botões.

Um grupo de botões a ser utilizado para a entrada de dados deve prever botões de rádio quando o conjunto de valores possíveis para um dado forem conhecidos, não excederem sete alternativas e forem mutuamente exclusivos. Se a escolha for múltipla o grupo deve prever botões de seleção.

8.2.3.4 - A lista de seleção

Uma lista de seleção corresponde a um campo de seleção multi-linear que abriga itens tipográficos, gráficos e mistos para uma escolha simples ou múltipla. Elas são empregadas para a entrada de dados cujos valores possíveis sejam conhecidos e em seu conjunto excedam as oito alternativas. O tamanho usual de uma lista deve permitir a visualização imediata de 7+- 2 itens. O projetista deve especificar a ativação de mecanismos de navegação internos (barras de rolamento) quando o número de escolhas possíveis se torne elevado. O limite máximo é de 50 itens que devem ser ordenados, iniciando-se pelas alternativas mais prováveis. Os separadores devem ser empregados para marcar a organização dos itens segundo grupos lógicos. Se qualquer organização não for possível, os separadores devem ser dispostos a cada 6 linhas de alternativas. Eles podem ser definidos como linhas em branco, de traços cheios ou pontilhados.

No caso de restrições de espaço elas podem ser configuradas como listas desdobráveis. Desta forma somente a primeira linha da lista é apresentada inicialmente. Para visualizar as outras alternativas o usuário deve acionar o botão de desdobramento que irá apresentar o restante da lista em modo de sobre-apresentação (primeiro plano da tela). O comprimento das linhas pode ser determinado pelo item mais longo se ele for menor do que 20 caracteres. Caso contrário ele deve ser definido pela média dos comprimentos de cada item.

Assim como para os botões, os itens desativados ou selecionados devem possuir uma forma coerente e diferenciável. A desativação pode ser representada através da cor cinza dos caracteres. Para os itens em foco o padrão é o vídeo reverso. O emprego de pequenos símbolos (✓, * ou flechas) pode ser especificado para marcar os itens já selecionados nas listas com múltiplas escolhas. Toda a lista de seleção deve possuir um rótulo identificativo a ser posicionado acima dela e alinhado pela esquerda. Se a lista for de desdobrar este rótulo deve permanecer à esquerda e alinhado pela sua margem inferior.

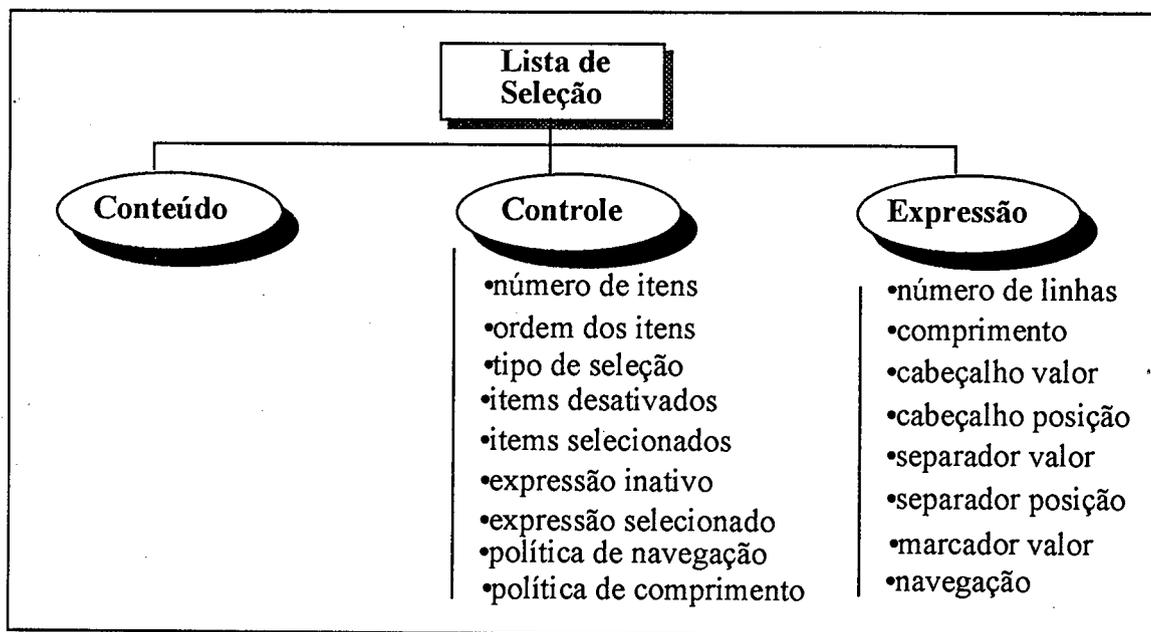


Fig 8.19 - A classe "Lista de Seleção"

8.3 - Os Comandos

A classe de Comandos é uma especialização da classe de Controles cujos elementos abrigam chamadas para as ações da aplicação. Em consequência deste parentesco direto com os controles, as classes de comandos são definidas segundo seus recursos de edição, seleção e de manipulação.

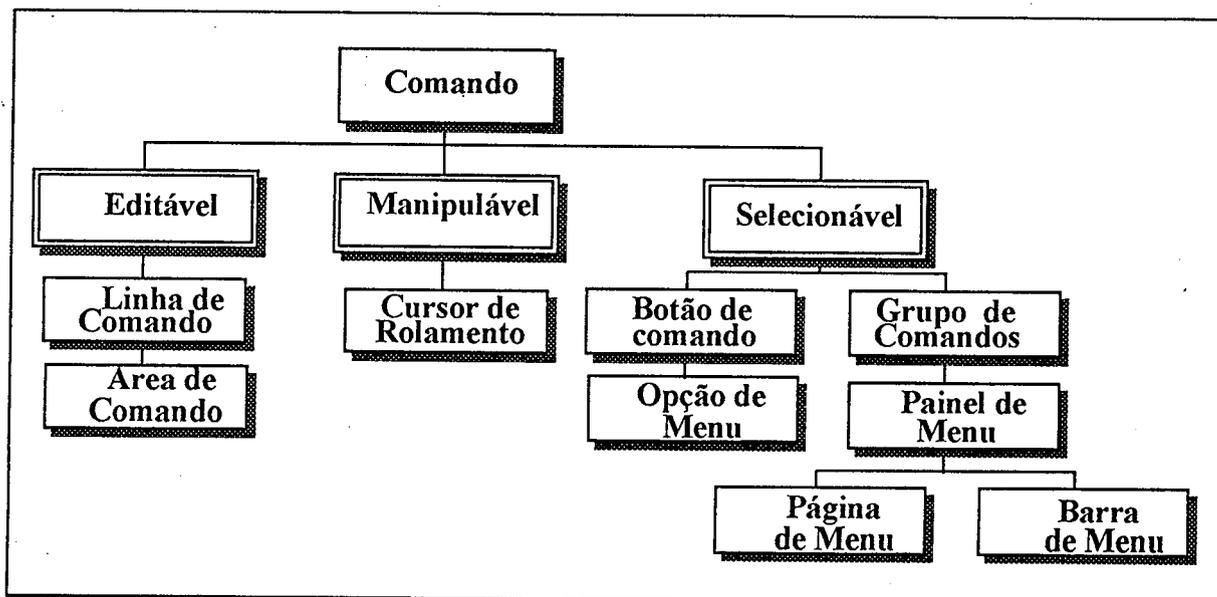


Fig 8.20 - As classes de Comandos

Os exemplos disponíveis em termos de comandos manipuláveis correspondem ao "cursor de rolamento" de uma barra de rolamento e às "marcas" de tabulação e margens de uma régua de edição de textos. O usuário age sobre o cursor para deslocar um documento sob uma janela. Sobre as marcas ele define diretamente as posições de tabulação e das margens de um parágrafo. As recomendações ergonômicas não tecem maiores comentários sobre a utilização deste tipo de comando.

8.3.1 - Os comandos editáveis: linha e área de comando

As áreas e linhas de comando correspondem a campos de edição uni e multi-lineares cujo conteúdo aciona uma funcionalidade do aplicativo. Neste sentido eles proporcionam grande flexibilidade ao usuário que pode se valer de todas as opções de vocabulário previstas numa linguagem de comando. A área de comando em especial fornece recursos de histórico para que o usuário possa avaliar e retomar estratégias de interação. Os comandos deste tipo devem estar localizados na parte inferior da tela e a área de comando não deve ser inferior à quatro linhas.

8.3.2 - Os comandos selecionáveis

Em relação aos comandos editáveis, os selecionáveis facilitam consideravelmente a tarefa do usuário, que realiza uma atividade mental de reconhecimento e não de recuperação. Nesta última o seu rendimento é muito superior. Outro aspecto importante dos comandos selecionáveis está ligado a abstração dos detalhes da sintaxe dos comandos. O usuário é conduzido na entrada de parâmetros de acordo com a opção de comando selecionada e os parâmetros já entrados. Este fato proporciona uma redução importante de erros de sintaxe.

8.3.2.1 - O botão de comando

O botão de comando é um campo para a seleção de comandos. Seus estados possíveis incluem "ativo", "inativo", "armado" e "ativado". O estado "armado" se instala quando o botão é definido como opção "par défaut" em um grupo de comandos. Nesta situação seu acionamento pode se dar tanto pelo mouse como através das teclas "Enter" ou "Return". Os nomes de botões que acionam caixas de diálogo devem ser seguidos de pontos de continuação

"...".

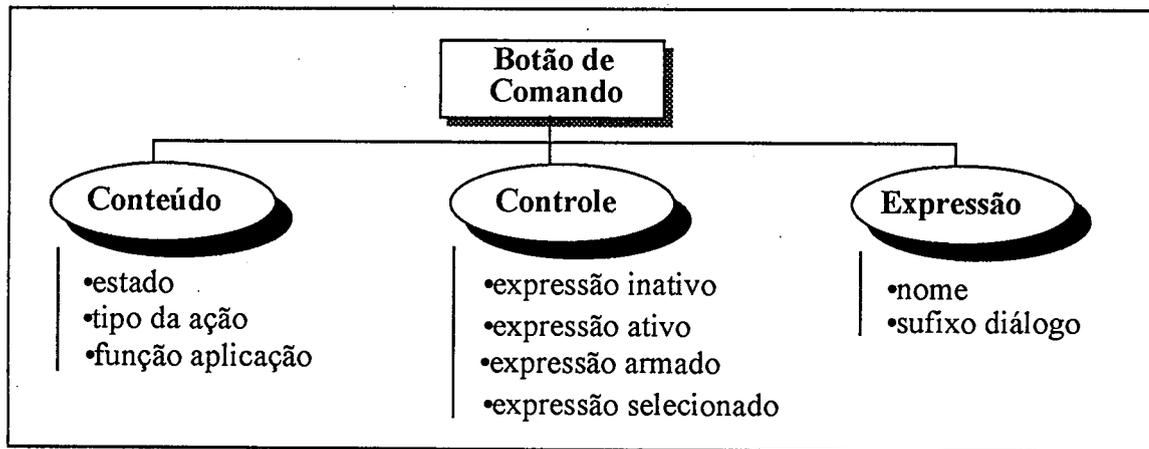


Fig 8.21 - A classe "Botão de comando"

8.3.2.2 - A opção de menu

A opção de menu corresponde a um campo uni-linear preenchido com o nome correto de um comando do sistema. Este é enviado para a interpretação pelo núcleo funcional assim que a opção seja selecionada pelo usuário. As opções de menu não ocorrem isoladas, pois são definidas como integrantes de um todo chamado menu.

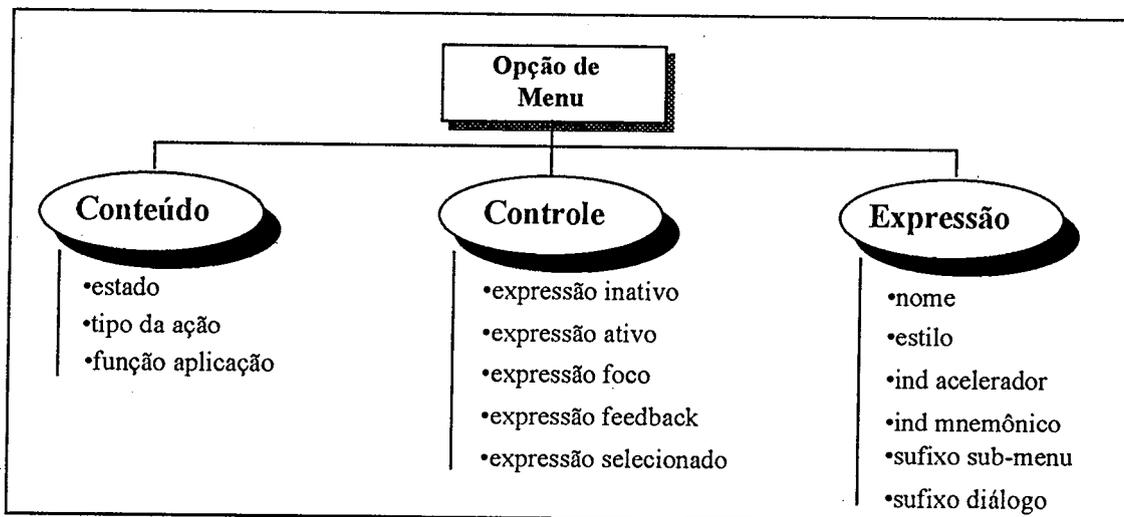


Fig 8.22 - A classe "Opção de Menu"

Elas podem ser classificadas como: opção de comando, opção de diálogo e opção de sub-menu. As opções do primeiro tipo acionam um comando da aplicação. As de diálogo acionam a apresentação de uma caixa de diálogo para precisar os parâmetros do comando. As de sub-menu causam a apresentação de um outro menu, o que permite ao usuário afinar a sua escolha em termos de opções de comandos. Estes diferentes tipos de opções devem ser

apresentados diferentemente. As opções que acionam caixas de diálogo devem ser seguidos de pontos de continuação "...". Aquelas associadas à sub-menus podem apresentar um marcador em forma de flecha, alinhado pela margem direita do menu.

As opções de menu podem assumir os estados "inativo", "ativo", em "foco", em "seleção" e "selecionado". Estas devem ser adequadamente representadas.

A seleção pode ser feita através do mouse ou do teclado. Na interação por mouse a área de seleção deve ser suficientemente ampla. A interação por teclado possibilita dois tipos de atalhos; os aceleradores e os mnemônicos. Um acelerador é um dispositivo que permite acionar uma opção de menu a partir de uma tecla de função ou de uma combinação de teclas. Isto ocorre independentemente da opção estar ou não visível. Neste caso a indicação da tecla associada deve fazer parte do rótulo da opção, geralmente à direita do nome da opção. Os mnemônicos acionam opções de menu que estejam visíveis. Trata-se de uma combinação de teclas envolvendo ou a letra inicial da opção ou uma outra que lhe seja marcante. Os mnemônicos devem estar representados pelo sublinhamento da letra definida no nome da opção. Seja qual for o meio de interação, somente a primeira letra do nome da opção deve estar em maiúsculo.

As opções podem assumir formas gráficas, tipográficas ou mistas. No caso de palavras, as denominações devem começar por um verbo, usual e simples, e devem evitar as abreviações. As opções de menu inativas devem estar diferenciadas coerentemente, isto é, de uma maneira dissimulada.

8.3.2.3 - O grupo de comandos

O grupo de botões de comando deve ser definido nas situações em que as opções de comandos possíveis forem em número reduzido. Seu arranjo pode seguir duas regras. Eles podem ser alinhados verticalmente e à direita do objeto a que fazem referência, ou horizontalmente e abaixo deste objeto. Um botão "par défaut" deve ser declarado e posicionado ao alto, se a orientação for vertical, ou à esquerda, no caso de uma orientação horizontal.

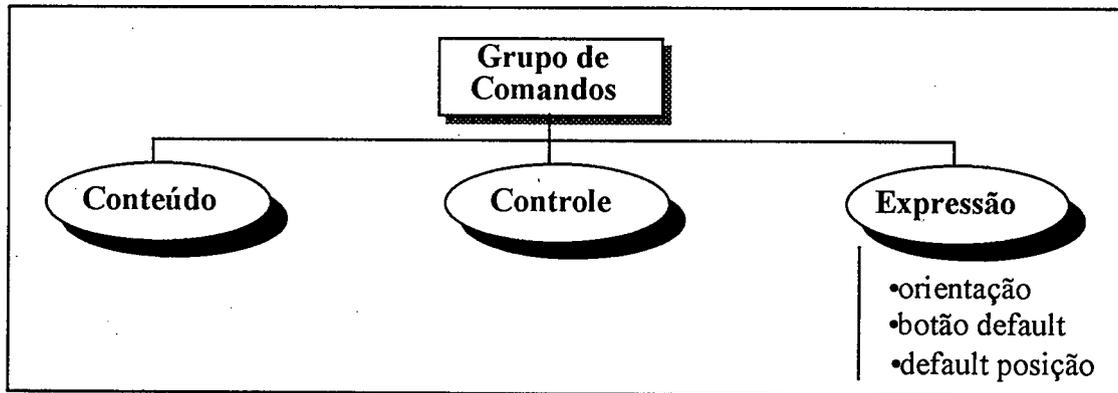


Fig 8.23 - A classe "Grupo de Comandos"

8.3.2.4 - O painel de menu

Um painel de menu corresponde a uma lista de seleção cujos itens são opções de menu ligadas a comandos, a caixas de diálogo e a outros menus. Em um menu as escolhas só podem ser simples. Todas as opções de um menu devem ser apresentadas simultaneamente, não sendo recomendados os dispositivos de navegação.

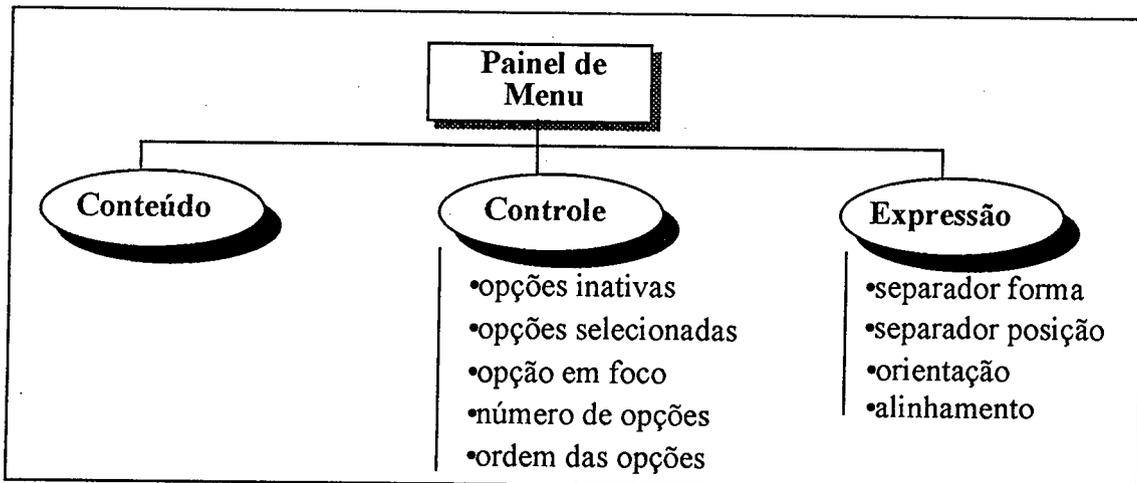


Fig 8.25 - A classe "Painel de Menu"

As opções devem refletir as necessidades da tarefa, devendo ser desprezadas as não pertinentes. Elas devem ser arranjadas segundo a ordem lógica da tarefa ou segundo a frequência de uso. Os menus muito numerosos devem ser organizados em grupos separados por um traço cheio ou uma linha em branco. As opções inativas não podem ser alvo de seleção. A orientação preferencial de um menu deve ser a vertical. A orientação horizontal deve ser considerada quando o número e o comprimento das opções for reduzido, como no caso da barra de menu.

Um painel de menu pode ser estático ou dinâmico. Os estáticos permanecem fixos em uma determinada posição da tela. Os dinâmicos só aparecem nela enquanto o usuário efetua uma seleção. Os menus dinâmicos, "de desdobrar" (pull-down), de "cascata" (cascading) e os de "sobre-apresentação" (pop-up) se diferenciam devido à forma de acionamento. O menu de desdobrar é apresentado na tela através do acionamento de uma opção de menu principal. O menu em cascata designa um menu de desdobrar que é apresentado a partir do acionamento de um outro menu de desdobrar. O surgimento na tela de um menu em sobre-apresentação (pop-up) independe de outra opção de menu. Ele ocorre simplesmente através do acionamento de um determinado botão do mouse. As opções de um menu pop-up variam conforme a posição do cursor na tela no momento em que ocorreu o acionamento. O painel de menu dinâmico deve ser apresentado próximo do local onde seu acionamento ocorreu. Os itens mais frequentes devem ocupar as primeiras linhas e sua orientação deve ser vertical com alinhamento pela esquerda.

Sob o ponto de vista da tarefa, os painéis podem ser principais e secundários. O menu principal representa o ponto de partida para a tarefa. Suas opções devem cobrir todas as alternativas em termos de ações básicas. Os menus secundários são acionados através de uma opção de menu e permitem ao usuário afinar a escolha de um comando.

Sob o ponto de vista do raciocínio por regras, os painéis de menus apresentam duas especializações; as páginas de menu e as barras de menu.

8.3.2.4.1 - A página de menu

A página de menu é característica dos chamados menus arborescentes. Trata-se de uma estrutura de encadeamento de sucessivas páginas de menus com objetivo típico de recuperação da informação em grandes bases de dados. Os aspectos principais destas estruturas de menus correspondem à largura (número de alternativas por página) e à profundidade (número de páginas por busca).

Uma página de menu deve possuir um título significativo, convite à interação (prompt), numeração e um histórico descritivo dos passos percorridos na interação.

O alinhamento do título e das opções de menus de página inteira deve ser definido pela esquerda. O Prompt deve ter a forma de um símbolo especial que lhe deve ser exclusivo e invariável. Sendo a seleção por teclado, as opções devem ser rotuladas com números e não

com letras. Esta numeração deve ser alinhada pela direita e separada do nome da opção através de um ou dois espaços. Ela deve partir sempre do número 1. No caso das letras serem definidas para a seleção por teclado, a numeração deve ser alinhada pela esquerda.

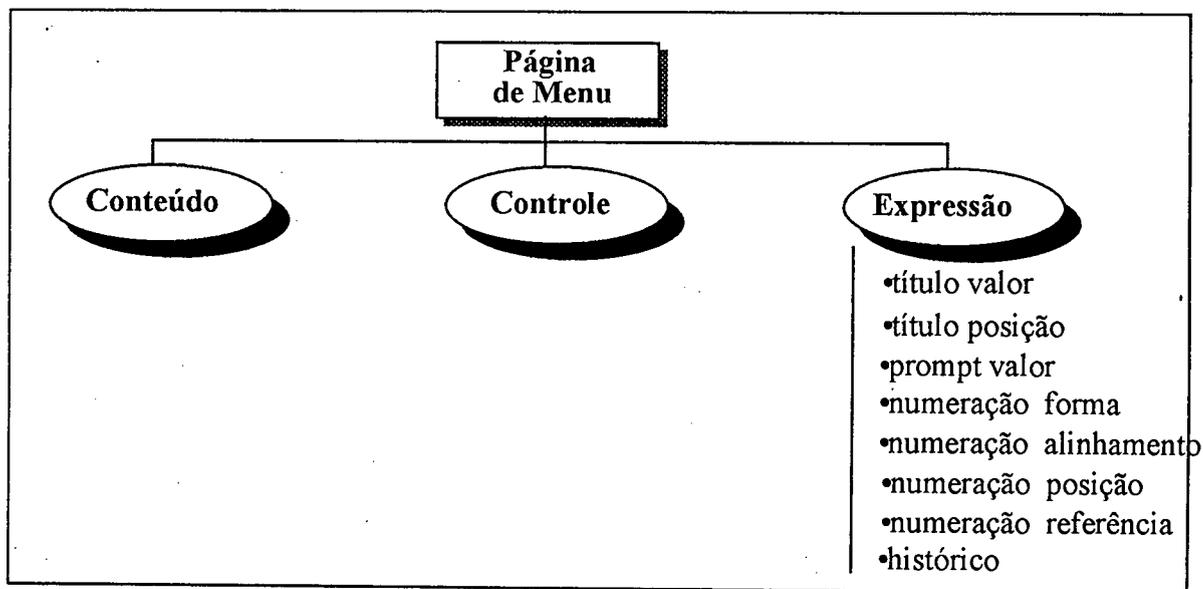


Fig 8.25 - A classe "Página de Menu"

8.3.2.4.2 - A barra de menu

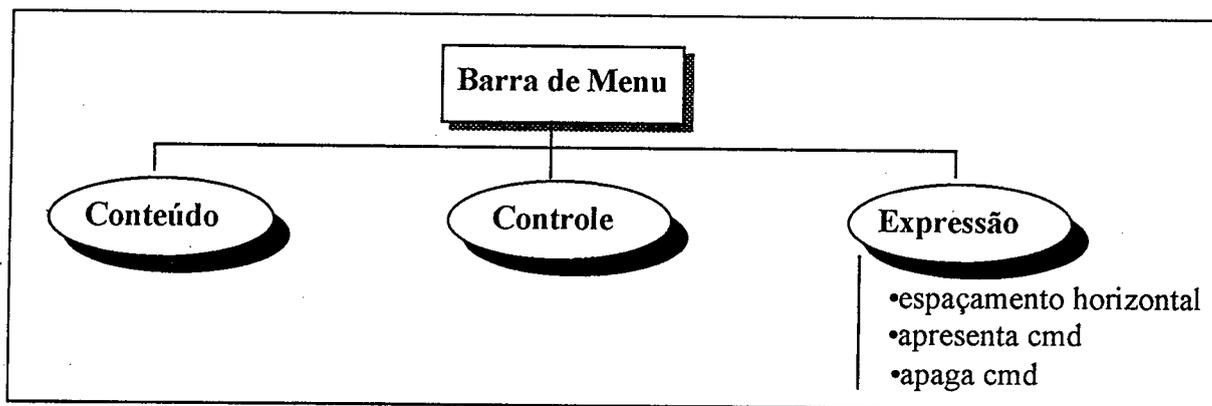


Fig 8.26 - A classe "Barra de Menu"

A barra menu tem orientação horizontal e deve ser posicionada no alto da tela ou da janela da aplicação. Ela apresenta os comandos de base de um sistema ou de um aplicativo que não devem ultrapassar as oito opções. Elas não comportam qualquer tipo de navegação e sua retirada da tela só se justifica para sistemas destinados ao grande público ou se expressamente comandada pelo usuário. Os nomes de suas opções devem ser palavras substantivas, não compostas e curtas, separadas por ao menos três espaços em branco. Se a ordem relativa à

tarefa ou à frequência de uso não podem ser definidas, um ordenamento alfabético deve ser empregado.

8.4 - Os Painéis de Controle

Os "Painéis de Controle" são OIA^e compostos que fornecem ao usuário um cenário adequado, em termos dos diferentes tipos de mostradores, de controles e de comandos necessários para a realização de sua ação ou tarefa.

Eles estão divididos em Janelas e Caixas de Diálogo. As janelas correspondem à expressão global de aplicativos e de documentos. As Caixas de diálogo fornecem informação e apoiam as tarefas e ações individuais.

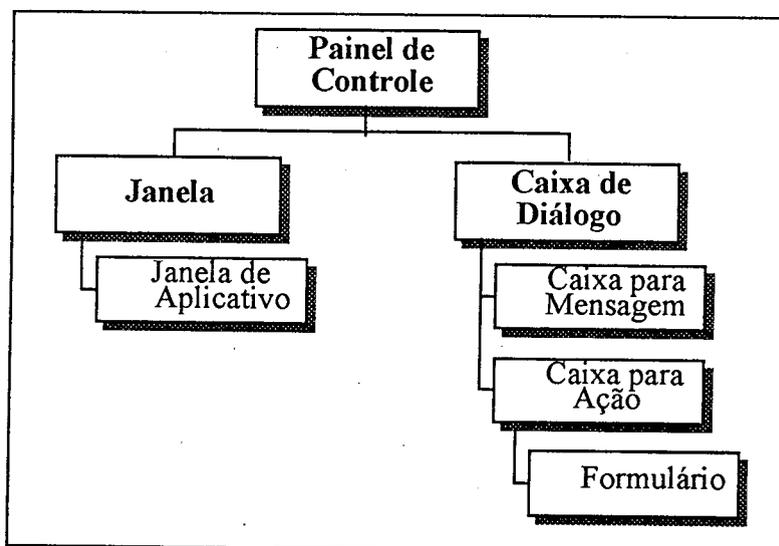


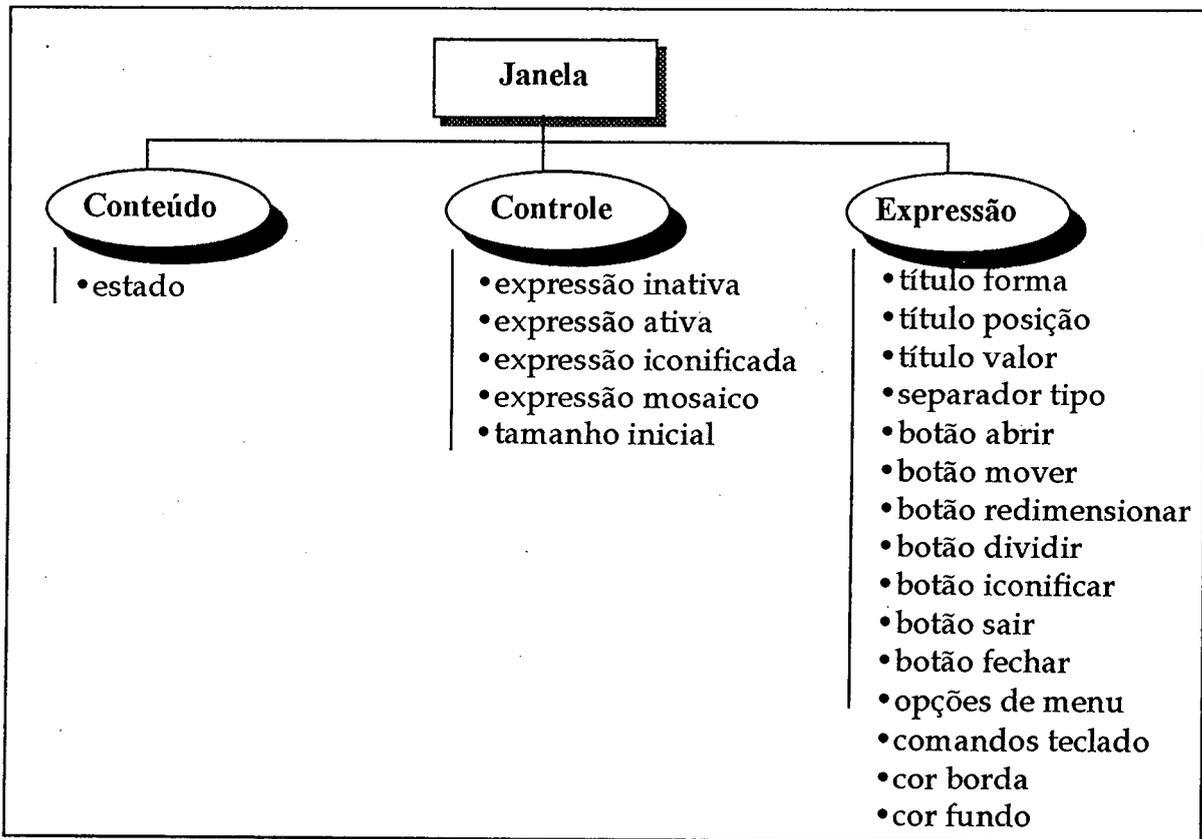
Fig 8.27 - As classes de Painéis de Controle

8.4.1 - A Janela

A janela gerada por um programa aplicativo pode assumir os estados "inativo", "ativo", "mosaico" e "iconificado". A cada um deles deve corresponder uma representação adequada. A janela ativa possui o foco das ações do usuário até que um outro foco seja definido. Ela é comumente colorida com cores densas. Os tons cinza são empregados para as janelas inativas. Na apresentação das janelas justapostas as barras de menu podem ser dispensadas.

Toda a janela deve possuir um título alfanumérico curto e significativo. Em caso de restrição de espaço, ele pode ser abreviado para que não ultrapasse os 4 caracteres. A posição do título deve ser mantida inalterada para todas as janelas do sistema. Estas devem propor

botões e/ou opções de menu e/ou comandos de teclado associados às ações de abertura, fechamento, de movimentação, de redimensionamento, de iconificação, de segmentação e de passagem para uma outra janela. As janelas de documentos devem apresentar barras de rolamento preferencialmente verticais. Nelas deve ser previsto os botões para a segmentação da janela. Esta separação deve ser indicada por uma linha cheia ou tracejada. A dimensão inicial proposta para uma janela deve ser compatível com as disponibilidades em termos de tela física, ficando à cargo do usuário o seu redimensionamento.



Fug 7.? - A classe Janela

8.4.1.1 - A janela de aplicativo.

A janela do aplicativo corresponde a uma área do terminal físico alocada para uma aplicação específica ou para o programa gerente de janelas (Finder, Presentation Manager, ...). Estes aplicativos podem por sua vez, criar diversas outras janelas de documentos que vão coexistir na tela. Em vista desta possibilidade, toda janela de aplicativo deve possuir uma opção de menu para o controle da disposição das janelas secundárias. Esta opção desdobrada deve incluir a lista dos documentos abertos e a lista de comandos relativos à disposição das janelas; justaposição, superposição, ou em cascata. A justaposição deve ser preferida quando as

manipulações sobre as janelas não forem frequentes ou quando o usuário for iniciante. Casos contrários, a opção preferível deve ser a superposição. A disposição em cascata pode tornar explícitas as interrelações de dependências entre janelas do mosaico. Aconselha-se estabelecer um limite para a quantidade de janelas abertas, que não deve ultrapassar as 7 janelas.

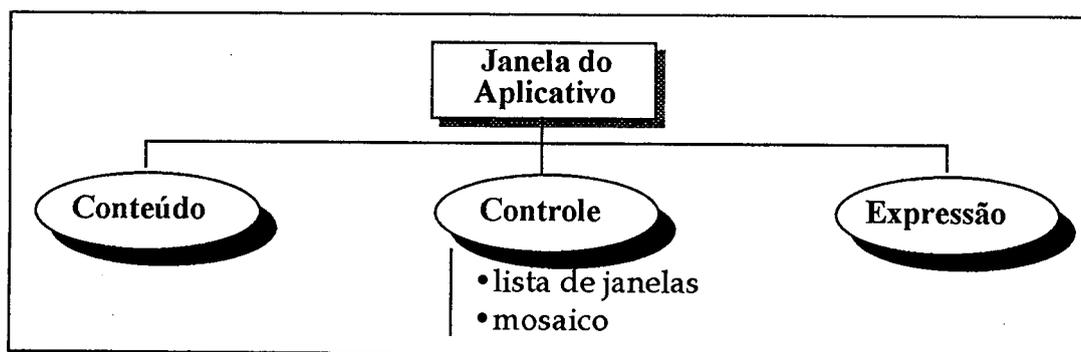


Fig 8.29 - A classe Janela do Aplicativo

8.4.2 - A caixa de diálogo

A caixa de diálogo corresponde a uma janela especialmente destinada à apresentação de mensagens e/ou de comandos para ações.

As caixas de diálogo podem ser modais ou não modais. Elas são modais quando exigem uma resposta do usuário, que fica impedido de qualquer outra ação até que isto aconteça. As não modais permitem que o usuário trabalhe sobre outros objetos de uma outra janela ou caixa de diálogo, enquanto que aguardam uma ação sua.

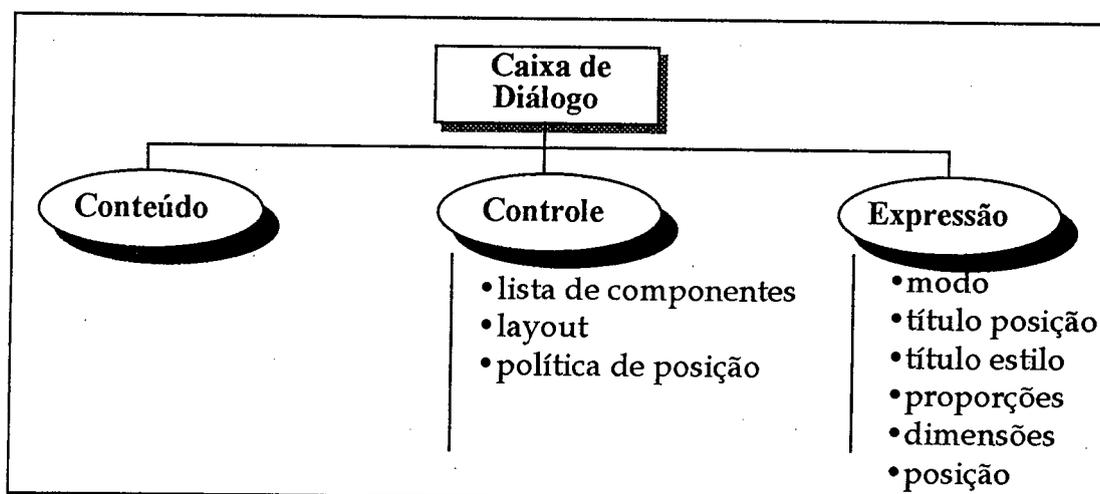


Fig 8.30 - A classe Caixa de Diálogo

As dimensões de uma caixa de diálogo devem ser definidas em função de seus componentes. A

relação entre o comprimento e altura deve se aproximar do chamado número de ouro; $1/1,618$. A política referente à sua aparição na tela pode ser de três tipos; padronizada, atraída pelo objeto a que se refere ou alinhada com o título da caixa de diálogo que lhe é superior. Seu título deve ser centrado na margem superior ou alinhado pela esquerda e a distribuição de seus integrantes pode seguir o desenho de uma grade. A densidade final de uma caixa de diálogo não pode exceder os 40 %.

8.4.2.1 - As caixas de mensagem

As caixas de mensagem fornecem informações e instruções ligadas à condução, à ajuda, às advertências, aos alarmes e aos erros na interação. Elas são do tipo modais, exigindo que o usuário tome seu conhecimento para permitir a sequência da interação. A mensagem deve ser adequada ao tipo de situação. Aconselha-se filtrar as mensagens provenientes do sistema operacional, do ambiente gráfico ou mesmo da aplicação e traduzi-las em termos da linguagem natural. O botão "ok" deve ser previsto em toda a caixa de mensagem como meio de receber a confirmação do usuário.

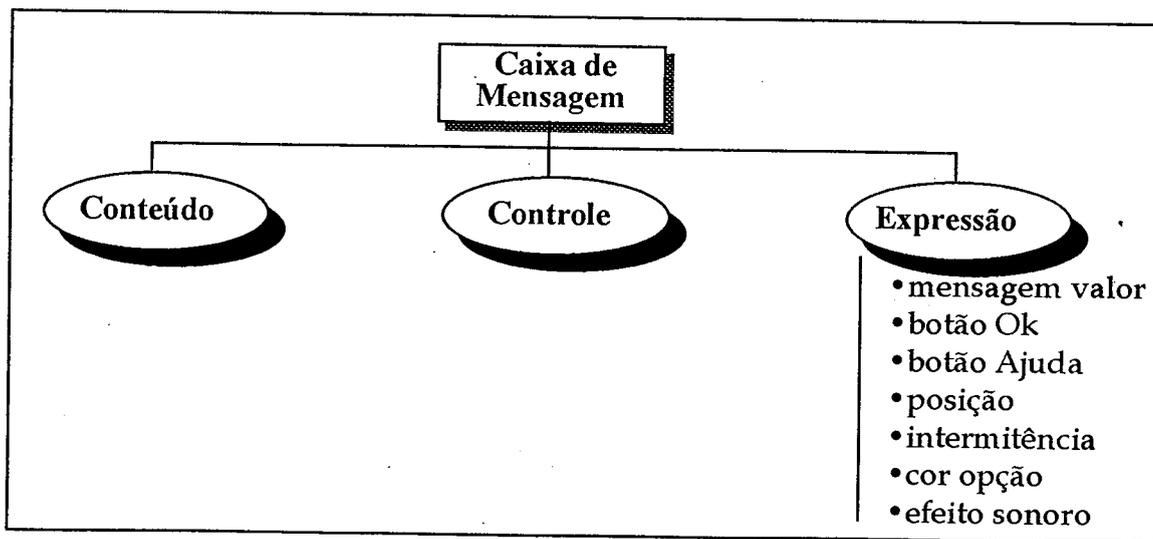


Fig 8.31 - A classe Caixa de Mensagem

Quando a situação for de alarme as caixas devem estar destacadas através de uma localização central da tela, do emprego da cor vermelho, da intermitência ou do acionamento de um efeito sonoro. Numa situação de advertência elas devem explorar o estereótipo do amarelo. As caixas de mensagens de erro devem propor sempre um botão "Ajuda".

8.4.2.2 - A caixa de Ação/Tarefa

A caixa de ação/tarefa proporciona os controles e comandos específicos para a introdução de parâmetros e para o acionamento da ação ou da tarefa que a define. Este tipo de caixa deve abrigar os botões de comando para validar, para aplicar imediatamente e para cancelar uma ação. Um deles deve ser definido "par default" e diferenciado apropriadamente. No caso de ações destrutivas, a opção "par default" deve recair sobre sua anulação e não sobre a própria ação. Também devem ser previstos botões de ajuda e de fechamento da caixa. A definição do layout destes botões deve minimizar os movimentos do mouse.

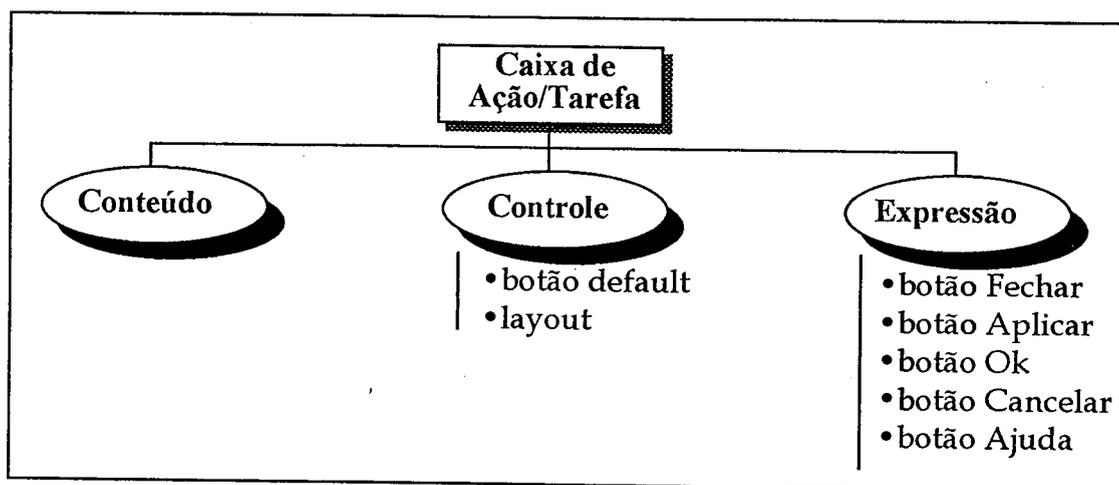


Fig 8.32 - A classe Caixa de Ação/Tarefa

8.4.2.3 - A caixa de Formulário

O formulário proporciona um painel para a apresentação e para a entrada de dados que apresenta um layout compatível com os documentos físicos manuseados pelo usuário em sua tarefa. A exemplo da classe de controles, uma caixa de formulário possui um mecanismo de proteção contra a modificação de dados considerados críticos para o sistema. Ao ser acionado este mecanismo torna o formulário insensível às ações do usuário.

O início das ações de entrada deve se dar a partir do campo localizado mais ao alto e à esquerda do painel. Esta posição inicial deve ser mantida de forma consistente durante a interação. Os campos devem estar adequadamente identificados, sendo que seus rótulos podem conter informação sobre as unidades e os valores aceitáveis para os dados a entrar. O alinhamento dos rótulos deve ser feito pela esquerda. Caso eles se apresentem comprimentos

muito diferentes seu alinhamento pode ser feito pela direita. O mecanismo de navegação entre os campos deve ser de simples operação. Aqueles contendo dados críticos para o sistema devem ser identificados e protegidos contra acidentes de operação. Um efeito sonoro pode ser empregado para informar sobre a proteção dos campos toda a vez em que o usuário tentar um acesso. Um formulário deve contar com a participação de mostradores de dados sobre o estado do sistema, linhas e até mesmo áreas de comandos. Estes devem estar localizados na parte inferior do painel. Esta mesma região deve ser reservada para a apresentação caixas de mensagens de erro. O registro dos dados entrados em um formulário não deve ser acionado como efeito colateral da ação de entrada de algum dado. O usuário deve explicitamente solicitar este registro através de um botão de comando. Este deve ter uma denominação coerente. Aconselha-se que seu rótulo contenha a palavra "Entre". Da mesma forma deve ser previsto um botão para cancelar este comando. Após o registro das entradas o sistema deve preencher com traços "----" os campos desconsiderados pelo usuário. O alinhamento dos dados numéricos deve ser feito a partir do ponto decimal.

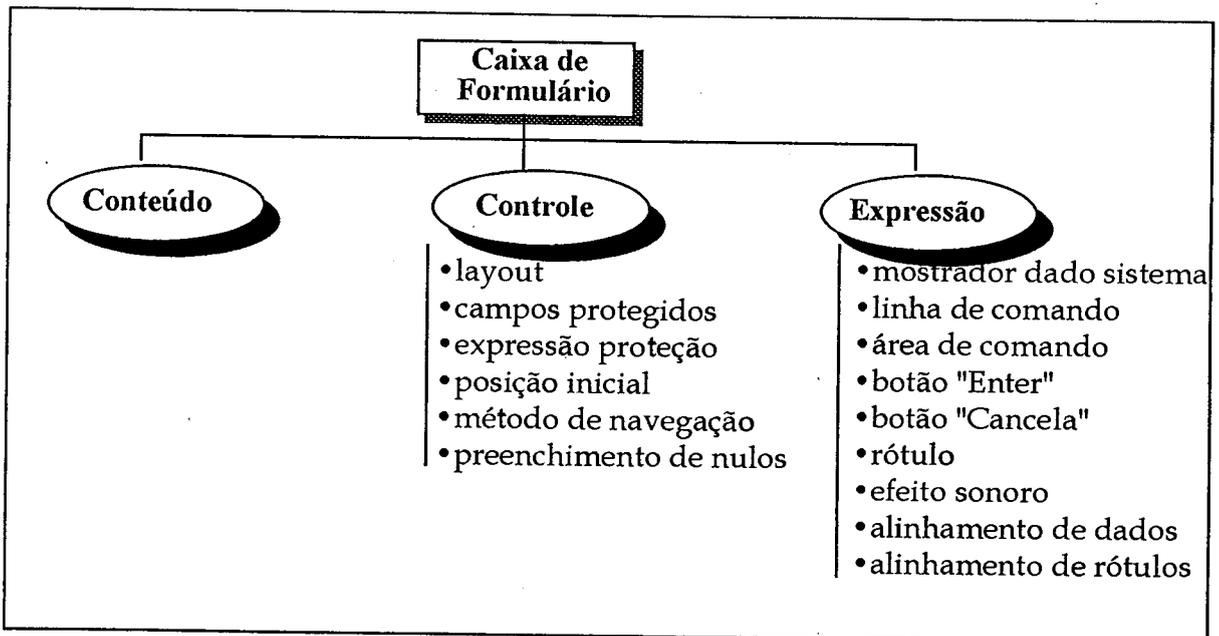


Fig 8.33 - A classe Caixa de Formulário

8.5 - O resumo do capítulo

✓ A definição de classes de funções baseia-se na interação dos OIA^e com o usuário e com a aplicação. Os exemplares destas classes modelam os componentes de uma ferramenta genérica; mostradores, controles, comandos e painéis de controle.

✓ Os mostradores são interadores que apresentam a capacidade de "acionar" o usuário através da apresentação das diferentes classes de dados e de informações de que ele necessita para pensar sua tarefa. Por outro lado eles são acionados por um programa aplicativo, que define quando, como e com quais valores os mostradores devem ser propostos ao usuário.

✓ Os controles são OIA^e sensíveis às ações do usuário, lhe proporcionando facilidades em termos de edição, seleção e manipulação direta. Esta classe de objetos não têm os recursos para acionar uma funcionalidade da aplicação. Assim, ela possibilita a entrada de dados que servirão como parâmetros para a ativação posterior de um comando.

✓ A classe de Comandos é uma especialização da classe de Controles cujos elementos abrigam chamadas para as ações da aplicação.

✓ Os "Painéis de Controle" são OIA^e compostos que fornecem ao usuário um cenário adequado, em termos dos diferentes tipos de mostradores, de controles e de comandos necessários para a realização de sua ação ou tarefa.

9 - DISCUSSÕES

Esta monografia se encerra com as discussões referentes as propostas apresentadas e as perspectivas em termos de pesquisas futuras ligadas aos objetos de interação abstratos ergonômicos.

9.1 - O resumo das propostas apresentadas.

A problemática geral deste trabalho refere-se à consideração efetiva dos preceitos da ergonomia cognitiva nas decisões de projeto de interfaces homem-computador. A forma de apoio pretendida refere-se a um ambiente computacional para a geração de interfaces ergonômicas, cuja base de regras é derivada das recomendações ergonômicas.

A identificação do conjunto de objetos de interface para o raciocínio ergonômico representa o objetivo específico deste trabalho. Face ao grande número de ambientes de programação e de objetos neles disponíveis, a primeira decisão se impõe. Quais objetos devem compor o ambiente de apoio pretendido? A resposta aponta para os objetos de interação portáteis ou abstratos, definidos, seja a partir do conjunto união ou intersecção dos objetos de interação dos diferentes ambientes.

9.1.1 - OIA^e como facilitador do raciocínio ergonômico

Verifica-se entretanto a impossibilidade de estabelecer uma correspondência efetiva entre as recomendações ergonômicas e um conjunto de objetos e de atributos abstratos. De fato, estas recomendações se referem a componentes de interface que existem nos modelos mentais dos ergonomistas e não em um ambiente de concepção. É necessário então definir um outro conjunto de objetos de interação, cujo nível de abstração se aproxime do verificado nas recomendações ergonômicas. Os OIA^e objetos de interação abstratos ergonômicos e seus atributos têm suas origens nas próprias recomendações. Eles são abstratos pois estão desvinculados de qualquer ambiente de concepção possível e são ergonômicos pois são derivados do exame de recomendações ergonômicas.

Apesar delas serem repletas de indicações sobre classes e atributos de componentes de interface, as recomendações ergonômicas nada indicam entretanto, sobre como organizar estes componentes, seja a nível de atributo seja a nível de classe. Estas definições estruturais são de

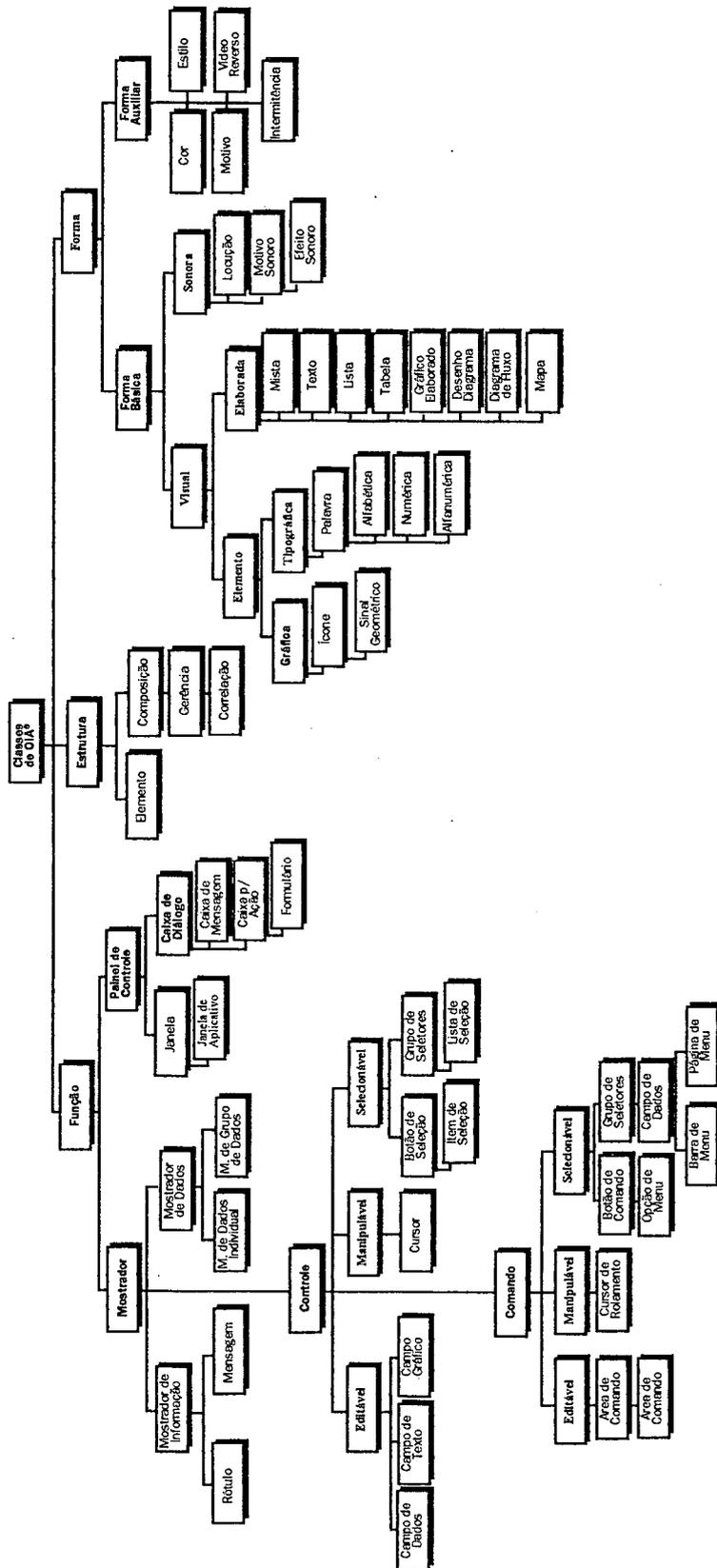


Fig 9.1 - As classes de objetos de interação abstratos ergonômicos - OIA^o -

fundamental importância para alcançar o êxito pretendido na identificação dos objetos de interface e de seus atributos ergonômicos.

Um OIA^e apresenta a morfologia tridimensional conteúdo-controle-expressão. Características em dimensões diferentes aumentam as possibilidades de correspondências com as recomendações ergonômicas. Se a maior parte destas recomendações refere-se aos aspectos da apresentação dos objetos, são em bom número também, aquelas que referem-se aos conteúdos a veicular, e à maneira pela qual realizar esta codificação. Um OIA^e atrai de uma maneira efetiva boa parte das recomendações relativas aos aspectos da sintaxe e da semântica de um sinal.

Outra característica explorada na definição das classes de OIA^e refere-se à repartição da semântica de um sistema interativo entre seus componentes, em especial entre os objetos interativos. Sob o ponto de vista de engenharia de software uma repartição de funções alivia o núcleo funcional e diminui a frequência de trocas entre componentes. Em consequência do esvaziamento do núcleo funcional ocorre o fortalecimento semântico dos OIA^e. Os conteúdos sugeridos nos textos das recomendações incluem o estado (ativo, inativo, em foco, em seleção, selecionado, default), a conformidade (ok, não usual, desvio, erro) e a implicação (alarme, anomalia, incidente). Verificar as conformidades, informar sobre as implicações e sobre as disponibilidades são ações típicas de elementos de interface. A identificação e a incorporação desta semântica pelos OIA^e favorece sua correspondência com as recomendações ergonômicas. Estes componentes encontram-se em um nível de abstração alguns degraus acima do verificado com os objetos de interação atuais.

Um OIA^e define um componente de interface cuja origem nas recomendações ergonômicas, cujas três dimensões de atributos e cujo elevado nível de abstração possibilitam uma correspondência efetiva entre suas características configuráveis e as recomendações ergonômicas. Sua aplicação em um ambiente lógico de apoio em ergonomia torna direto e descomplicado o raciocínio baseado em regras.

9.1.2 - OIA^e como recurso para a usabilidade

As classes e atributos de OIA^e apresentadas na segunda parte desta monografia representam os recursos de que dispõe o projetista para configurar a usabilidade de suas interfaces. Eles foram identificados no exame de cerca de 700 recomendações ergonômicas de

diversos autores e encontram-se distribuídos segundo classes de recursos funcionais, estruturais e formais.

Os recursos de formas básicas e auxiliares são definidos a partir de uma relação de determinação/subordinação. O mesmo ocorre entre estruturas elementares e compostas. Já as funções são definidas na metáfora dos componentes de uma ferramenta genérica informatizada: mostradores, controles, comandos e painéis de controle.

9.1.3 - OIA^e como sinal computacional

As soluções propostas para as questões de estrutura do modelo estão influenciadas pela perspectiva semiótica. Um sistema interativo é visto como um sistema de sinais, através dos quais o usuário aciona e interpreta os objetos de sua tarefa. Da combinação das contribuições da semiótica com técnicas de engenharia de software e das próprias recomendações ergonômicas, surge a proposta de OIA^e como sinal computacional. Um OIA^e representa a primitiva de uma possível programação orientada a sinais.

O paradigma de programação orientado a objetos se justifica na analogia com o mundo não informatizado, que é repleto de objetos. Ele representa uma ruptura na tradição, ao propor entidades englobando ao mesmo tempo a estrutura de dados e as ações possíveis sobre ela. Uma possível programação orientada a sinais se justifica na própria natureza de um sistema interativo, repleto de sinais.

A proposta da programação orientada a sinais é feita por Andersen (Andersen, 1990). Na analogia com a orientação à objetos...

"... cada sinal é caracterizado por atributos e por uma sequência de ações, muitas das quais o usuário pode ver ou escutar (os efeitos), mas algumas delas ele deve inferir. Os sinais organizam a substância de expressão gerada pela execução de um programa, e as transformações nesta substância são reflexo das ações dos sinais."

Este autor no entanto, não avança em sua proposta, que depende fundamentalmente da definição das primitivas de uma possível linguagem orientada a sinais...

"...Uma pré-condição para a construção de tal linguagem deveria estar no estabelecimento de um conjunto de taxemas através da análise dos sinais computacionais existentes."

A proposta dos OIA^e representa um avanço nesta direção, que é feito entretanto, à partir do exame das recomendações ergonômicas. Ela explora as noções de sinal e de agente PAC para compor uma entidade lógica com o objetivo básico de explicitar conteúdos, expressões e suas interações. Os sinais, assim como as recomendações ergonômicas estão ligados à mesma lógica; a utilização do sistema.

A um sinal, mais do que a um objeto, interessa os recursos básicos para ser destacado, discriminado e associado em um painel. Sua configuração pode ocorrer em dois planos, o do conteúdo e o da representação. Independentemente da aplicação, o projetista pode definir as expressões correspondendo aos diversos conteúdos. Pode-se mesmo pensar em componentes padronizados que tenham expressões destaque, associação e discriminação sugeridas "de fábrica". Isto garante uma uniformidade na expressão que vai contribuir para a consistência do projeto. Um segundo nível de configuração refere-se aos conteúdos que devem ser transmitidos aos usuários. Neste nível o projetista preocupa-se exclusivamente com as listas de objetos a destacar, a discriminar e a associar, despreocupando-se de como representar estas relações na tela.

9.2 - Pesquisas futuras

As frentes de trabalho abertas a partir da definição das classes de OIA^e incluem diversas atividades de pesquisa e desenvolvimento. As listadas no seguimento desta conclusão correspondem às atividades para as quais já se conta com a colaboração de pesquisadores bolsistas de mestrado do Curso de Pós-Graduação em Engenharia de Produção e Sistemas da Universidade de Santa Catarina.

9.2.1 - A validação do modelo de OIA^e

O tipo de validação já iniciada neste trabalho corresponde à validação por adesão. Ela confunde-se com o próprio desenvolvimento do modelo na medida em que consiste em verificar a capacidade dos OIA^e em atrair recomendações ergonômicas. A etapa de definições estruturais envolveu cerca de 200 recomendações utilizadas como controle para as diversas estruturas propostas. Tendo sido definidas a morfologia e a tipologia dos OIA^e realizou-se uma etapa de identificação de exemplares de classes, de sub-classes e de atributos. Foram então consideradas mais de 500 recomendações de diversos autores que aderiram, sem dificuldades à estrutura proposta. Foi possível comprovar então que esta estrutura é

suficientemente geral e aberta para definição de novas classes e de novos atributos de OIA^e. Este tipo de validação vai prosseguir com a atividade prevista de ampliação da base de recomendações que aderem ao modelo de OIA^e.

O segundo tipo de validação corresponde a validação pelo raciocínio ergonômico. Trata-se de uma atividade de manipulação envolvendo grupos de projetistas que serão convidados a avaliar, sob o ponto de vista ergonômico, a interface de um aplicativo. O projeto desta manipulação prevê: (i) a definição da população, (ii) a proposta um método de análise que uniformize a população, (iii) a definição dos limites da análise (iv) a definição das etapas de testes e (v) a definição dos métodos de análise dos dados. Espera-se comprovar nestes testes a validade do modelo de OIA^e como facilitador de qualquer tipo de raciocínio ergonômico.

9.2.2 - O desenvolvimento do aplicativo HyperAttribut

HyperAttribut é um aplicativo interativo cujo objetivo é de apoiar não só as atividades de desenvolvimento do modelo de OIA^e, como também aquelas de especificação e de avaliação de uma interface homem-computador.

Suas funções principais visam o registro, a classificação e a recuperação das recomendações ergonômicas segundo as classes e os atributos dos OIA^e já definidas ou ainda inéditas. HyperAttribut possibilita tanto a ampliação da base de recomendações que aderem ao modelo como a definição de novas classes de OIA^e dentro do modelo. Ela é constituída de duas pilhas HyperCard/HyperTalk denominadas "Recomendações" e "Classes".

A pilha de recomendações proporciona a interface com o expert em ergonomia que realiza o registro das recomendações. As cartas desta pilha respeitam a anatomia de um OIA^e, na medida em que propõe campos para receber além do nome da classe, os nomes dos atributos de conteúdo, de controle e de expressão a que uma recomendação se refere. As funcionalidades inseridas nestas cartas se destinam especialmente à ampliação do modelo e da base de recomendações a ele acopladas. As principais são as seguintes;

- a importação de recomendações a partir de um arquivo texto com a consequente criação de uma nova carta na pilha.
- o auxílio à decisão sobre qual classe de objetos e quais atributos desta classe são os mais adequados para receberem a recomendação atual. As funções são realizadas por

campos de rolamento que recuperam as classes de objetos e os diferentes atributos já definidos no modelo. Um histórico das decisões já realizadas a partir de recomendações semelhantes à que está sendo tratada completa este conjunto de funcionalidades.

- o registro, isolado ou em lotes, das recomendações ergonômicas, se necessário com a criação de novas classes e atributos na pilha "Classes". Esta constitui a principal função deste módulo, e a ela podem ser associadas a que modifica e a que anula completamente dos dados referentes a uma recomendação já registrada.
- diversos tipos de navegação com funções de; busca a partir de uma expressão qualquer; acesso a todas as cartas definidas para uma mesma classe de objetos na pilha "Recomendações"; acesso a uma determinada classe, ao "Índice" e ao "Sumário" da pilha "Classes".

As cartas da pilha "Recomendações" abrigam também campos para a edição de informações adicionais sobre as recomendações, como a referência, os comentários, os exemplos e contra-exemplos de sua aplicação, etc. Estes campos representam o apoio ao entendimento das recomendações.

A pilha "Classes" representa o módulo de recuperação das recomendações ergonômicas. Esta pilha possui cartas de três tipos. As cartas "Classe" com as classes de objetos, uma carta "Índice" e uma carta "Sumário". Nas cartas "Classe" estão registrados os três tipos de atributos de um OIA^e com as respectivas recomendações. Suas funcionalidades permitem;

- o exame dos atributos da classe através de sucessivos menus pull-down. Cada sequência de examinação é acompanhada da apresentação das recomendações que dizem respeito a cada opção de menu. No caso de objetos compostos o exame de seus atributos pode se estender a várias cartas dos componentes deste objeto. Neste caso, funções especiais de navegação são ativadas para que o examinador possa a qualquer instante retornar à classe do objeto composto inicial. No caso de sub-classes os atributos e recomendações herdadas de outras classes são igualmente apresentadas.
- o acesso às cartas que abrigam as recomendações mostradas na janela de informação que acompanha o exame dos atributos de uma classe. Esta funcionalidade tem o

objetivo de proporcionar ao examinador uma série de informações adicionais sobre a recomendação, como fonte, comentários, exemplos e contra-exemplos de sua aplicação; etc.

- a montagem de relatórios que, com diferentes níveis de detalhe, apresentam atributos, instâncias de atributos e recomendações já registradas em uma determinada classe.

Esta pilha propõe também funcionalidades voltadas para o desenvolvimento e para a manutenção do modelo e da base de recomendações. Elas são as seguintes;

- a montagem das heranças de novas classes em função das já existentes no modelo.
- a realização de modificações dos nomes dos objetos e dos atributos com a necessária repercussão sobre todo o sistema aplicativo.

Este aplicativo constitui a maquete de uma ferramenta de ajuda à especificação e à avaliação das interfaces com o usuário baseada no modelo de OIA^e. Ele oferece o acesso à informação ergonômica e um tipo de condução por objetos para as ações de especificação e de avaliação das interfaces homem-computador.

Pretende-se o desenvolvimento de uma nova versão de HyperAttribut em plataformas diversas (Macintosh, PC e estações de trabalho). A definição desta nova versão resultará da análise do desempenho da maquete existente nos testes de validação do modelo de OIA^e pelo raciocínio ergonômico. A idéia é de aproveitar a mesma manipulação para verificar a facilidade de uso e a adaptação à tarefa (utilisabilidade) da versão atual de HyperAttribut. Os dados desta manipulação definirão as características de sua nova versão, que será distribuída em shareware.

9.2.3 - A ampliação e o refinamento da base de recomendações ergonômicas

A pilha de recomendações ergonômicas registradas em HyperAttribut se aproxima atualmente das 700 cartas. Pretende-se a realização de uma etapa de trabalhos visando (i) a ampliação deste número, (ii) a tradução dos textos das recomendações para a língua portuguesa e (iii) o refinamento desta base pelo esclarecimento sobre as recomendações conflitantes.

9.2.4 - A ligação utilização/funcionamento

O confronto entre as lógicas de utilização e de funcionamento das interfaces homem-computador é o objeto de exame de uma atividade de pesquisa futura. Esta prevê a implementação de recursos de OIA^e e de suas correspondências com os objetos concretos.

A partir das classes de componentes funcionais pretende-se a construção de recursos de OIA^e e a montagem de uma "toolbox abstrata". Sua utilização por um grupo de projetistas deverá ser o objeto de um terceiro tipo de validação do modelo; a validação pelo uso. Ela se baseia na possibilidade de especificar uma interface completa através das classes de OIA^e identificadas.

Isto realizado, o próximo passo deve examinar as correspondências entre os OIA^e da toolbox abstrata com os objetos concretos de diferentes toolboxes. Uma estratégia a ser examinada consiste na utilização dos objetos de interação abstratos como elo de ligação entre os OIA^e e os objetos concretos.

9.2.5 - Em busca de uma nova dimensão de OIA^e ; os sequenciais

Os aspectos sequenciais de uma interface homem-computador podem ser configurados através dos chamados esqueletos de aplicação. Eles definem segmentos de diálogo que são propostos "prets à employer" aos projetistas de interfaces. Pode-se aplicar à tarefa de identificação dos objetos sequenciais e de seus atributos ergonômicos a mesma estratégia de ação seguida para os objetos estáticos. A hipótese é de se poder definir diálogo como um sinal, de morfologia tridimensional e cujas classes são definidas a partir das perspectivas funcional, estrutural e formal.

10 - Referências Bibliográficas

- Andersen, P. B. (1990). A Theory of Computer Semiotics: semiotic approaches to construction and assessment of computer systems. (First ed.). Cambridge, U.K.: Press Syndicate of the University of Cambridge.
- Andersen, P. B. (1990b). Towards an aesthetics of hypertext systems: a semiotic approach. (Venus Report No. 3). Departement of Information and Media Science - University of Aarhus.
- Andersen, P. B. (1993). The force dynamics of interctive systems: Towards a computer semiotics. pp. 36. Department of Information and Media Science - Universtity of Aarhus.
- Aschehoug, F., & Scapin, D. L. (1990). Base de recommandations ergonomiques pour l'évaluation et la conception d'interfaces utilisateurs (Document Interne No. INRIA - Institut National de Recherche en Informatique et en Automatique).
- Baar, D. d., Foley, J. D., & Mullet, K. E. (1992). Coupling application design and user interface design. In INTERCHI'92 - Human Factors in Computing Systems, 1 (pp. 259-266). Monterey, CA: ACM Press.
- Barthet, M.-F. (1988). Logiciels interactifs et ergonomie: modèles et méthodes de conception (first ed.). Paris: Bordas.
- Bass, L., & Coutaz, J. (1991). Developing software for the user interface (first ed.). Reading, Massachusetts: Addison-Wesley Publishing Company.
- Bastien, C. & Scapin, D. (1993). Human factors criteria, principles, and recommandations for HCI: methodological and standatdisation issues. (Internal Repport). INRIA
- Bastien, C. J. (1991). Validation des critères ergonomiques pour l'évaluation d'interfaces utilisateurs. (Rapport de Recherche No. 1427). INRIA.
- Blattner, M. M., Sumikawa, D. A., & Greenberg, R. M. (1989). Earcons and Icons: Their Structure and Common Design Principles. Human-Computer Interaction, 4(1), 11-44.

- Bodart, F., & Vanderdonckt, J. (1993a). Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection. In A. Wesley (Ed.), INTERCHI'93 - Human Factors in Computing Systems, 1 (pp. 424 - 429). Amsterdam: ACM Press.
- Bodart, F., & Vanderdonckt, J. (1993b). Guide Ergonomique de la présentation des applications hautement interactives. (1° ed.). Namur, Belgique: Presses Universitaires de Namur.
- Brown, C. M. (1988). Human-computer interface design guidelines (first ed.). Norwood, New Jersey: Ablex Publishing Corporation.
- Card, S., Moran, T. & Newell, A. (1983). The psychology of human-computer interaction (first ed.). Hillsdale: Lawrence Erlbaum Associates.
- Chapanis, A. & B., W. (1990). Specifying human-computer interface requirements. Behaviour & Information Technology, 9(6), 479-492.
- Chauvet, J.-M., Harman, D., & Jouhier, B. (1991). Open Interface: un outil de construction d'interfaces graphiques portables. Genie Logiciel & Systèmes Experts, 24, 90-102.
- Chevalier, J. (1980). Dictionnaire des symboles (first ed.). Paris: Robert Laffont.
- Coutaz, J. (1990). Interfaces homme-ordinateur: conception et réalisation. (first ed.). Paris: Bordas.
- Cybis, W. (1990). Abordagem ergonômica para a concepção/avaliação de um sistema de informação. In Congresso Internacional da Tecnologia do Software, Telemática e Informação, (pp. 1-16). São Paulo:
- Fairchild, K., Meredith, G., & Wexelblat, A. (1989). A Formal Structure for Automatic Icons. Interacting with Computers, 1(2), 131-140.
- Faveaux, L., & Yedid, C. (1991). Réflexion sur la mise en oeuvre d'une démarche ergonomique de conception d'interface. In Y. Quéinnec & F. Daniellou (Ed.), Congress of International Ergonomics Association, 1 (pp. 649-651). Paris: Taylor & Francis.
- Foley, J. D. & V. D., A. (1984). Fundamentals of interactive computer graphics (first ed.). Massachusetts: Addison-Wesley Publishing Company.

- Gould, J. D., Boies, S. J., & Lewis, C. (1991). Making usable, useful, productivity-enhancing computer applications. Communications of the ACM, 34(1), 74-85.
- Gray, P. D., Waite, K. W., & Draper, S. W. (1990). Do-It-Yourself Iconic Displays: Reconfigurable Iconic Representations of Application Objects. In Proceedings of IFIP INTERACT'90: Human-Computer Interaction (pp. 639-644).
- Green, A. J. K., & Barnard, P. J. (1990). Iconic Interfacing: The Role of ICON Distinctiveness and Fixed or Variable Screen Locations. In Proceedings of IFIP INTERACT'90: Human-Computer Interaction (pp. 457-462).
- Guiraud, P. (1983). La sémiologie (4^o ed.). Paris: Presses Universitaires de France.
- Hollnagel, E. (1989). Information and reasoning in intelligent decisions support systems. In E. Hollnagel Mancini, G. & Woods, D. (Eds.), Cognitive engineering in complex dynamic London: Academic Press.
- ISO 9241 Part 10 (1993). Ergonomic requirements for office work with visual display terminals, Part 10 Dialogue Principles; Committee Draft ISO 9241-10
- ISO 9241 Part 14 (1993). Ergonomic requirements for office work with visual display terminals, Part 14 Menu Dialogues; Draft International Standard ISO 9241-14
- IBM (1989). Systems Application Architecture, Common User Access: Advanced Interface Design Guide. In Boca Raton: International Business Machines Corp.
- Johnson, J. (1992). Selectors, going beyond user-interface widgets. In INTERCHI'92 - Human Factors in Computing Systems, 1 (pp. 273-279). Monterey, CA: ACM Press.
- Johnson, J., Roberts, T., Verplank, W., Smith, D., Irby, C., Beard, M., & Mackey, K. (1989). The Xerox Star: a retrospective. IEEE Computer, 22(9), 11-29.
- Laville, A. (1977). Ergonomia (First ed.). São Paulo: Editora Pedagógica e Universitária Ltda.
- Loshe, G., Walker, N., Biolsi, K., & Rueter, H. (1991). Classifying Graphical INFORMATION. Behaviour and INFORMATION Technology, 10(5), 419-436.
- Morris, C. (1974). Fondements de la théorie des signes. Langages, 8(35), 15-21.

- Nadin, M. (1988). Interface design and evaluation - Semiotic implications. In H. R. Hartson & D. Hix (Eds.), Advances in human-computer interaction (pp. 376). Norwood, New Jersey: Ablex Publishing corporation.
- Norman, D. A. (1984). Cognitive engineering principles in the design of human-computer interfaces. In G. Salvendy (Eds.), Human Computer Interaction Amsterdam: Elsevier Science Publisher.
- Norman, D. A. (1991). Cognitive Artifacts. In J. M. Carrol (Eds.), Designing Interaction, psychology at the human-computer interface (pp. 17-38). Cambridge: Cambridge University Press.
- OSF (1990). Motif Style Guide (1.1 ed.). Cambridge, MA: Open Software Foundation.
- Microsoft Corporation(1987). Microsoft Windows: Software development kit: Application Style Guide (version 2).
- Pollier, A. (1991). Evaluation d'une interface par des ergonomes: diagnostiques et stratégies (Rapport de Recherche No. 1391). INRIA.
- Prieto., L. J. (1972). Messages et Signaux (first ed.). Paris: Presses Universitaires de France.
- Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs and symbols, and other distinctions in human performance models. IEEE Transactions on Systems, Man, and Cybernetics, SMC 13(3), 257-266.
- Ravden, S., & Johnson, G. (1989). Evaluating usability of human-computer interfaces (first ed.). Chichester: Ellis Horwood Limited.
- Richard, J.-F. (1983). Logique du fonctionnement et logique d'utilisation (Rapport de Recherche No. 202). INRIA.
- Richard, J.-F., Bonnet, C., & Ghiglione, R. (1990). Traité de Psychologie Cognitive : perception, action, langage(première ed.). Paris: Bordas.
- Rivlin, C., Lewis, R., & Davies-Cooper, R. (1990). Guidelines for Screen Design (first ed.). Cambridge: Blackwell scientif Publications.

- Rogers, Y. (1989). Icons at the Interface: Their Usefulness. Interacting with Computers, 1(1), 105-117.
- Sacre, B., Sacre-Provot, I., & Vanderdonckt, J. (1992). Une description orientée objet des objets interactifs abstraits utilisés en interfaces homme-machine. (rapport IHM/Ergo No. 10). Facultés Universitaires Notre-Dame de La Paix - Institut d'Informatique - Projet Trident.
- Scapin, D. L. (1986). Guide ergonomique de conception des interfaces homme-machine. (Rapport de Recherche No. 77). INRIA - Rocquencourt - France.
- Scapin, D.L. et al. (1988a) - La concéption ergonomique d'interfaces: problèmes de méthodes, Rapports de Recherche , Rocquencourt, France: INRIA.
- Scapin, D.L. (1988b) - Vers des outils formels de description des tâches, orienté conception d'interfaces, Rapports de Recherche , Rocquencourt, France: INRIA.
- Scapin, D.L. (1989a) - Guidelines for user-interface design: Knowledge collection and Organisation, Technical Raport TR.D12.1 Version Date 30/12/1989, Rocquencourt, France: INRIA.
- Scapin, D. L. (1989b) - MAD: Une méthode analytique de description des tâches. IN: Colloque sur l'engièrie des interfaces homme-machine, Sophia-Antipolis, France, INRIA
- Scapin, D. L. (1990a) - Aiding mechanisms for the design of user interface: Some researche issues. IN: Proceedings of the First International Conference on Automation Technologie. p 587-593, Taiwan.
- Scapin, D.L. (1990b) - Decyphering human-factors recommendations, IN: Karwoski, W. & Rahimi, M. (Eds). Ergonomics of hybrid automated Systems II, Amsterdam: Elsevier Science Publisher.
- Scapin, D.L. (1990c) - Des critères ergonomiques pour l'évaluation et la concéption d'interfaces. IN: Actes du Congres de la SELF, Montréal.
- Scapin, D. L. (1990d). Organizing human factors knowledge for the evaluation and design of interfaces. International Journal of Human-Computer Interaction, 2(3), 203-229.

- Sebillotte, S. (1991). Decrire des tâches selon les objectifs des operateurs: de l'interview à la formalisation (Rapport de Recherche No. 125). INRIA.
- Shneiderman, B. (1987). Designing the User Interface: strategies for effective human-computer interaction (first ed.). Addison-Wesley Publishing Company.
- Smith, S. (1986). Standards versus guidelines for designing user interface software. Behaviour and Information Technology, 5(1), 47-61.
- Smith, S. L., & Mosier, J. N. (1986). Guidelines for designing user interface software No. ESD-TR-86-278). The MITRE corporation.
- Sousa, F. d., & Bevan, N. (1990). The use of guidelines in menu interface design: evaluation fo a draft standard. In D. Diaper (Ed.), Human Computer Interaction - INTERACT 90, (pp. 435-440). Elsevier Science Publishers.
- Valentin, A., & Luongsang, R. (1987). L'ergonomie des logiciels (first ed.). Paris: ANACT.
- Ware, C. (1992). The foundations of experimental semiotics: a theory of sensory and conventional representation. Experimental Semiotics(6), 1-13.
- Williges, B., & Williges, R. (1984). Dialogue design considerations for interactive computer systems. In F. A. Muckler (Eds.), Human Factors Review Santa Monica: The Human Factors Society, Inc.
- Winograd, T., & Flores, F. (1986). Understanding computers and cognition: a new foundation for design (first ed.). Norwwod, New Jersey: Ablex Publishing Corporation.
- Zanden, B. V., & Myers, B. A. (1990). Automatic, look-and-feel independent dialog creation for graphical user interfaces. In (pp. 27-34). ACM Press.