

# PUC

---

Série: Monografias em Ciência da Computação,  
No.16/90

**INTERFACES INTELIGENTES:  
CONCEITUAÇÃO, TAXONOMIA E ARQUITETURA DE SUPORTE**

Regina H. B. Cabral

Departamento de Informática

---

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO  
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP-22453  
RIO DE JANEIRO - BRASIL

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Departamento de Informática

Série: Monografias em Ciência da Computação, Nº 16/90

Editor: Paulo A. S. Veloso

Novembro, 1990

**Interfaces Inteligentes: Conceituação,  
Taxonomia e Arquitetura de Suporte**

por

Regina Helena Bastos Cabral

DISCIPLINA : INF 3004 – Exame de Qualificação

ORIENTADORA : Profa. Dra. Clarisse Sieckenius de Souza

Outubro de 1989

**Responsável por publicações:**

Rosene Teles Lins Castilho  
Assessoria de Biblioteca, Documentação e Informação  
PUC RIO, Departamento de Informática  
Rua Marquês de São Vicente, 225 - Gávea  
22453 - Rio de Janeiro, RJ  
BRASIL

Tel.: (021) 529-9386  
BITNET: [userrtlc@lncc.bitnet](mailto:userrtlc@lncc.bitnet)

TELEX: 31078

FAX: (021) 274-4546

# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Fontes de conhecimento</b>	<b>3</b>
2.1	Conhecimento sobre o usuário . . . . .	3
2.1.1	Modelo de usuário . . . . .	4
2.1.2	Modelo mental . . . . .	6
2.1.3	Modelo do modelo mental . . . . .	8
2.2	Conhecimento sobre o domínio do problema . . . . .	8
2.3	Conhecimento sobre a interface . . . . .	8
2.4	Análise crítica . . . . .	9
<b>3</b>	<b>Tipos de diálogo</b>	<b>11</b>
3.1	O aspecto funcional . . . . .	11
3.2	O aspecto linguístico . . . . .	11
3.2.1	A forma da linguagem . . . . .	12
3.2.2	O acesso a dados e a objetos . . . . .	15
3.3	Análise crítica . . . . .	16
<b>4</b>	<b>Gerência da interface</b>	<b>18</b>
<b>5</b>	<b>Uma arquitetura</b>	<b>20</b>
<b>6</b>	<b>Conclusões</b>	<b>22</b>
	<b>Referências</b>	<b>25</b>

# Interfaces Inteligentes: Conceituação, Taxonomia e Arquitetura de Suporte

Regina Helena Bastos Cabral

Outubro 1989

## Resumo

O trabalho pretende identificar os componentes necessários à caracterização de uma interface como inteligente. Através da análise da literatura corrente sobre interfaces, são focalizados os tipos de diálogo e as fontes de conhecimento aos quais a interface precisa ter acesso, e são discutidos aspectos relevantes sobre sua gerência. São também propostas uma arquitetura mínima para interfaces inteligentes e uma taxonomia, ainda incompleta, dos vários sistemas e abordagens encontrados na literatura.

**Palavras-chave:** Interfaces Inteligentes, Modelagem de Usuário, Tipos de Diálogo, Arquiteturas de Interface.

## Abstract

This work attempts to identify the components needed to characterize an interface as intelligent. It focusses on the dialogue types and on the sources of knowledge needed by an interface, while discussing some relevant aspects on user interface management, through an analysis of the current literature. Furthermore, it proposes a minimal architecture for intelligent interfaces and a taxonomy, albeit incomplete, of systems and approaches found in the literature.

**Keywords:** Intelligent User Interfaces, User Modelling, Dialogue Types, User Interface Architectures.

# 1 Introdução

Interface pessoa-computador é o suporte de *software* e *hardware* através do qual ocorre um diálogo entre estes dois atores. O diálogo é a troca observável, entre pessoa e computador, de símbolos e ações [HART89].

Para caracterizar interfaces inteligentes é necessário definir os conceitos de interface e de inteligência. A definição de inteligência em sentido amplo é uma questão filosófica não resolvida. Entretanto, alguns aspectos aceitos como determinantes de inteligência são a capacidade de raciocínio, a habilidade de adquirir, aplicar e transmitir conhecimento, e a capacidade de adaptação a novas situações [WINS84], [WENG87], [RICH88], [HANC89]. Não há consenso na literatura sobre uma definição de interfaces inteligentes, conforme se verifica comparando, por exemplo, [RISS84], [BRAJ86], [BAEC87], [HANC89], [SZEK89].

Interfaces podem ser analisadas como tendo sido projetadas sob as seguintes quatro perspectivas diferentes [KAMM88]:

1. perspectiva do sistema: a interface é projetada com os objetivos de acelerar a transmissão de dados entre a pessoa e o computador, e de reduzir a possibilidade de erros.
2. perspectiva da ferramenta: a interface é projetada com o objetivo de facilitar a aplicação de ferramentas e avaliação dos resultados correspondentes por parte do usuário da aplicação.
3. perspectiva do meio de comunicação: a interface é projetada com o objetivo de facilitar a comunicação entre duas ou mais pessoas.
4. perspectiva do parceiro do diálogo: a interface é projetada com o objetivo de fazer com que o computador se comporte como se fosse uma pessoa agindo cooperativamente no processo de comunicação com o usuário.

Cada uma destas perspectivas pode atender a um ou mais aspectos citados como determinantes de inteligência. Por exemplo, as perspectivas do sistema e da ferramenta consideram a capacidade de adquirir, aplicar e propagar dados; a perspectiva do meio de comunicação considera a capacidade de transmitir dados; e a perspectiva do parceiro do diálogo supõe que a interface atenda a todos os aspectos citados como determinantes de inteligência. Não obstante, cada uma delas pode ser válida e adequada para auxiliar a alcançar objetivos diferentes e, portanto, não se pode dizer que uma seja melhor que outra. O importante é que uma interface pode e deve ser projetada usando uma combinação destas perspectivas de modo a resolver os problemas de comunicação, controle e utilização do sistema (interface e programa de aplicação) [RISS84], [CARD89], [CHIG89], [SZEK89].

O problema de comunicação se refere às decisões de *quais* informações devem ser transmitidas ao usuário, *quando* e *como* deve ocorrer esta transmissão. O problema de controle focaliza a questão *quem faz o que e quando*. O problema de utilização do sistema se refere à possibilidade de acesso de modo útil a todas as suas capacidades.

A resolução de todos estes problemas depende diretamente da capacidade de expressão da interface, do conhecimento que a interface tem sobre o usuário, sobre o problema focalizado e sobre si própria, e da capacidade que a interface tem para analisar e combinar estes conhecimentos de modo a auxiliar o usuário na sua solução.

O texto objetiva contribuir para a caracterização de interfaces inteligentes, estando organizado como a seguir. Na Seção 2 são discutidas fontes de conhecimento [RISS84] básicas para a interface tomar decisões de comunicação. Na Seção 3 são discutidos os tipos de diálogo através dos quais a interface pode efetivar a comunicação entre o usuário e o programa de aplicação (no que se segue, referido como *processador de tarefas*). Na Seção 4 são focalizados aspectos relativos a gerência de interfaces. Na Seção 5 é sugerida uma arquitetura mínima para interfaces inteligentes. Na Seção 6 são apresentadas conclusões e perspectivas para pesquisa futura. No final do texto são apresentadas as referências bibliográficas.

## 2 Fontes de conhecimento

As fontes de conhecimento [RISS84] às quais uma interface inteligente precisa ter acesso podem ser classificadas em fontes de conhecimento sobre o usuário, sobre o problema e sobre a própria interface. Características importantes a considerar sobre usuário e interface serão amplamente discutidas neste texto. O conhecimento sobre o problema é específico de cada domínio de aplicação: neste texto, serão apenas citados aspectos gerais que devem ser considerados no projeto de interfaces.

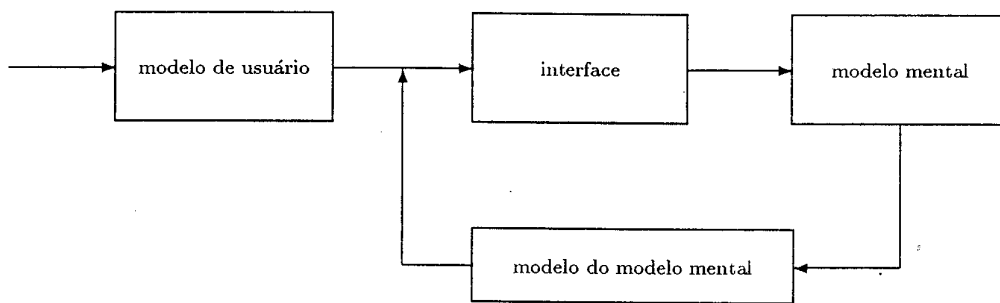
### 2.1 Conhecimento sobre o usuário

É consenso na literatura que a qualidade da interface depende da atenção que ela dá a algumas características do usuário, entre as quais destacam-se [RICH83], [RISS84], [GOOD84], [AIR85a], [AIR85b], [APPE85], [SHNE87], [CARD89]:

- seu grau de experiência com sistemas de computação;
- sua visão sobre a linguagem e terminologia de comunicação com a interface;
- sua atitude em relação à interface, isto é, se é cooperativa, desconfiada, confusa, etc.
- seu interesse em conhecer mais sobre o sistema e em adquirir a capacidade de usá-lo sem auxílio;
- sua visão sobre o domínio do problema, incluindo modelos do mundo e mecanismos de raciocínio;
- seus objetivos a serem alcançados através do sistema;
- o histórico de seu desempenho durante o uso da interface.

A combinação destas características induz a construção de uma modelagem de usuário. A modelagem de usuário pode ser vista em três níveis, a saber, o modelo de usuário, o modelo mental e o modelo do modelo mental [RICH83], [APPE85], [MORA87], [NORM87], [CHIG89]. O modelo de usuário define um estereótipo [RICH83] de usuário. O modelo mental é o que o usuário adquire sobre o sistema durante sua utilização. O modelo do modelo mental é o modelo inferido da observação e análise do comportamento do usuário durante o uso do sistema.

A interface ideal é alcançada através de um processo iterativo e iterativo conforme ilustra a Figura 2.1. Inicialmente, é criado um modelo de usuário a partir do qual a interface é projetada e implementada. Ao usar esta interface, o usuário adquire seu modelo mental. Pela observação e análise do comportamento do usuário durante a utilização do sistema, é criado o modelo do modelo mental. Este último modelo é usado para adequar a interface, permitindo que o usuário atualize seu modelo mental e assim por diante.



*Figura 2.1 Modelagem de usuário*

### 2.1.1 Modelo de usuário

A consideração de um estereótipo de usuário pode servir como ponto de partida para o projeto da interface [RICH83]. Pesquisas em diferentes áreas procuram classificar as pessoas conforme suas características atemporais, como habilidades, experiências, motivações, personalidade e estilo de trabalho. Estas classificações são interessantes para direcionar o projetista na definição dos objetivos da interface, e reforçam a necessidade do trabalho conjunto de profissionais de diversas áreas para o projeto de interfaces.

Pesquisas em ergonomia, sociologia e psicologia focalizam características físicas, culturais e de personalidade do usuário, num espectro que inclui sexo, idade, estatura, sensibilidade da visão a cores e luminosidade, sentido de leitura (por exemplo, esquerda para direita), origem étnica, dentre outros, [GAIN86], [BAEC87], [KANT89], [EBER89]. Resultados destas pesquisas influenciam o projeto físico e semântico dos locais e equipamentos de trabalho.

Em [SHNE87], por exemplo, apresenta-se uma classificação dos diferentes tipos de personalidade baseada na teoria de tipos de personalidade formulada pelo psicólogo suíço Carl Jung. A classificação é composta por quatro dicotomias, a saber:



1. extroversão/introversão: o extrovertido gosta de estímulos externos e novos; o introvertido prefere padrões familiares, permanecendo com suas próprias idéias e trabalhando sozinho.
2. sensibilidade/intuição: o sensível gosta de rotinas estabelecidas e de trabalhos minuciosos; o intuitivo gosta de resolver problemas novos e não gosta de perder tempo com detalhes.
3. percepção/avaliação: o perceptivo gosta de aprender sobre novas situações mas tem dificuldades em tomar decisões; o avaliador gosta de planejar cuidadosamente a forma de alcançar seus objetivos e tenta seguir seus planos mesmo que fatos novos mudem seus objetivos.
4. sentimento/raciocínio: o sentimental se preocupa com os sentimentos de outras pessoas, tenta ser agradável, e se relacionar bem com a maioria das pessoas; o racional trata as pessoas impessoalmente e gosta de ter lógica em tudo o que faz.

A classificação do usuário quanto ao grau de experiência com sistemas de computação é essencial para o projeto de interfaces. [SHNE87] classifica o usuário como novato, ocasional e experiente. O usuário novato é totalmente inexperiente com o sistema e desconhece a sintaxe de seu uso. Ele pode ainda ser inexperiente com o próprio computador e tender a deixar que sua insegurança iniba seu aprendizado. O usuário ocasional conhece mas usa esporadicamente o sistema (ou uma variedade de sistemas) de computação. Entretanto, ele tem dificuldade em memorizar a sintaxe de utilização do sistema e gostaria de apenas reconhecer os comandos. Já o usuário experiente é completamente familiarizado com os aspectos sintáticos e semânticos do sistema e quer suas tarefas executadas rapidamente.

Estas características dos usuários novatos, ocasionais e experientes indicam que a interface será por eles considerada amigável e eficiente [SHNE87]:

1. se, para o usuário novato, a interface tiver:
  - (a) o vocabulário restrito a um pequeno número de termos consistentes entre si, significativos e familiares ao usuário;
  - (b) *feedback* informativo sobre o acompanhamento de cada tarefa;
  - (c) mensagens de erro construtivas e específicas;
  - (d) um sistema de segurança contra ações perigosas;
  - (e) tutoriais em linha;
  - (f) manuais com tutoriais impressos altamente explicativos.

Todas estas características visam possibilitar ao usuário a execução de tarefas simples, aumentar sua confiança [MUIR87] no sistema, reduzir sua ansiedade e insegurança, e reforçar positivamente o seu sucesso.

2. se, para o usuário ocasional, a interface tiver:

- (a) linguagem de comandos com terminologia significativa, estrutura simples e consistente;
- (b) *feedback* informativo sobre o acompanhamento de suas tarefas mais importantes ou mais demoradas;
- (c) mensagens de erro significativas;
- (d) um sistema de segurança contra ações perigosas;
- (e) auxílio em linha significativo mas resumido;
- (f) manuais impressos.

Todas estas características visam facilitar ao usuário o reconhecimento dos comandos necessários à execução de suas tarefas e possibilitar que ele explore com segurança comandos novos ou parcialmente esquecidos.

3. se, para o usuário experiente, a interface tiver:

- (a) comandos que exijam pouca digitação ou seleção;
- (b) *feedback* breve e em pequeno número;
- (c) formas de acelerar a entrada de comandos tais como a definição de macros para abreviar sequências de comandos frequentemente usados;
- (d) tempo de resposta rápido;
- (e) manuais impressos.

Todas estas características visam possibilitar que o usuário execute rapidamente suas tarefas.

### 2.1.2 Modelo mental

O modelo mental é o modelo formado na mente do usuário enquanto ele utiliza o sistema [APPE85], [NORM84], [NORM87], [MORA87], [EBER89]. O usuário pode formar um modelo mental relativo aos estados do sistema e um modelo mental dos comandos que provocam a mudança de um estado para outro. Estes modelos permitem que o usuário entenda o sistema e interaja com ele.

Durante a fase de aprendizado, o usuário descobre que uma ação particular gera uma correspondente reação. À medida que o aprendizado prossegue via uso do sistema e análise do diálogo ação-reação, o usuário sintetiza um modelo que mapeia reação desejada e ação necessária. Este modelo corresponde ao modelo de comandos do sistema. Nesta fase, o usuário tenta aprender a sintaxe da linguagem de comandos do sistema e não todo o seu vocabulário.

O modelo relativo aos estados do sistema começa a ser adquirido na fase de aprendizado e continua a se formar até que o usuário tenha explorado o sistema em sua plenitude ou não seja mais capaz de deduzir o modelo pela exploração do sistema. Este modelo está completo quando o usuário conhece todo o vocabulário da linguagem de comandos do sistema.

Por um lado, a aquisição dos modelos mentais depende de fatores pessoais como base técnica, tipo de raciocínio e de processamento de informações do usuário, e de sua experiência com outros sistemas semelhantes. Por outro lado, a formação destes modelos é efetivada pela exploração do sistema. Esta exploração é feita principalmente via mudança de estado no sistema. Ela é encorajada e tem sucesso apenas se a interface apresenta alguma simetria permitindo, facilmente, a volta ao estado original. Ou seja, ela é encorajada se o usuário tem segurança de que uma ação incorreta não exigirá grande perda de tempo ou esforço mental para ser desfeita. Portanto, as características da interface e a qualidade dos materiais de instrução sobre seu uso são decisivos no processo de aquisição dos modelos mentais pelo usuário.

À medida em que o usuário adquire experiência com o sistema ele faz simplificações através de homomorfismos [MORA87]. Estes homomorfismos são mapeamentos de vários estados do sistema para um estado mais abstrato, possivelmente correspondente a um objetivo do usuário. Isto sugere algumas regras para o projeto de uma interface que auxilie o usuário na aquisição de seu modelo mental:

1. descobrir homomorfismos naturais para o sistema através de regras de decomposição [CONA76], [HIMM73];
2. construir a malha correspondente a estes homomorfismos e então projetar a interface com base nesta malha;
3. considerar os aspectos que foram perdidos em cada ponto desta malha e construir métodos de auxílio para que o usuário incorpore estes aspectos nos homomorfismos de mais alto nível;
4. proporcionar ao usuário formas eficientes para movimentação entre quaisquer nós da malha.

Baseado nas experiências envolvendo uma variedade de tarefas, pessoas e dispositivos de interação, [NORM87] apresenta as seguintes observações sobre modelos mentais:

- são incompletos – as pessoas por inexperiência ou por desinteresse conhecem apenas superficialmente o sistema em uso (características funcionais e de dispositivos).
- dependem da habilidade do usuário – dependem de sua experiência com sistemas e mecanismos semelhantes, de sua capacidade de percepção, de sua personalidade, etc.
- são instáveis – os usuários se esquecem de detalhes do sistema em uso, principalmente se aqueles detalhes (ou todo o sistema) não são usados com frequência.
- não têm fronteiras fixas – operações semelhantes podem ser confundidas.
- não são científicos – os usuários mantêm conscientemente padrões de comportamento desnecessários a fim de poupar esforço mental.

- são parcimoniosos – frequentemente, a fim de reduzir a complexidade mental, os usuários fazem operações físicas extras em vez de fazer um planejamento mental que evitaria aquelas ações. Isto é especialmente verdade quando as ações extras permitem uma regra simplificada que se aplica a vários dispositivos, minimizando possibilidades de confusão.

### 2.1.3 Modelo do modelo mental

O modelo do modelo mental é criado a partir da análise de experimentos e observações feitos sobre o usuário durante a utilização do sistema, ou seja, por inferências feitas a respeito do modelo mental adquirido pelo usuário. Ele corresponde ao aprimoramento do estereótipo definido no modelo de usuário e permite a reformulação da interface adequando-a às necessidades do usuário.

Uma abordagem comumente usada para auxiliar na obtenção deste modelo é a análise do histórico da interação obtido pela monitoração do comportamento real dos usuários durante o uso do sistema. A monitoração pode ser feita para classificar o usuário conforme seus erros [COOM84], [GOOD84], [CARV88], ou conforme o vocabulário que ele use [RICH83], [CARR87]. Na monitoração de erros é desejável que se obtenha não só uma taxonomia dos erros do usuário mas, principalmente, um entendimento teórico das causas destes erros.

A análise do vocabulário permite inferir o nível do usuário a partir de suas ações e respostas. Entretanto, a falta de precisão na linguagem normalmente usada pelas pessoas não garante que o usuário seja adequadamente classificado. Se por um lado um usuário novato provavelmente não coloca questões em um vocabulário muito técnico, por outro, o usuário experiente pode usar um vocabulário geral e ser classificado pelo sistema num nível elementar.

A complexidade e importância da correção desta análise exigem que ela seja amparada por teorias de cognição e psicologia do comportamento e aprendizado.

## 2.2 Conhecimento sobre o domínio do problema

Para haver comunicação entre dois agentes é essencial que ambos tenham conhecimento sobre os aspectos básicos do domínio do problema focalizado [RISS84], [BACHaa], [LITMaa], [ADAR84], [CHIG89], [SZEK89]. Entre estes aspectos, destacam-se:

- conceitos e descrições específicos do domínio;
- estruturas hierárquicas necessárias para compreensão do domínio;
- tarefas que podem ser efetivadas no domínio através do uso do sistema.

## 2.3 Conhecimento sobre a interface

Entre as informações que a interface precisa ter a respeito de si própria destacam-se [RISS84], [CROF84], [ADAR84], [ELKE89], [SZEK89]:

- funções disponíveis no processador de tarefas;
- linguagens e dispositivos de comunicação disponíveis;
- fontes de conhecimento às quais tem acesso, por exemplo, modelagem de usuário e do problema;
- quando e como se adaptar ao usuário;
- quando e como atribuir responsabilidades ao usuário ou assumir responsabilidades.

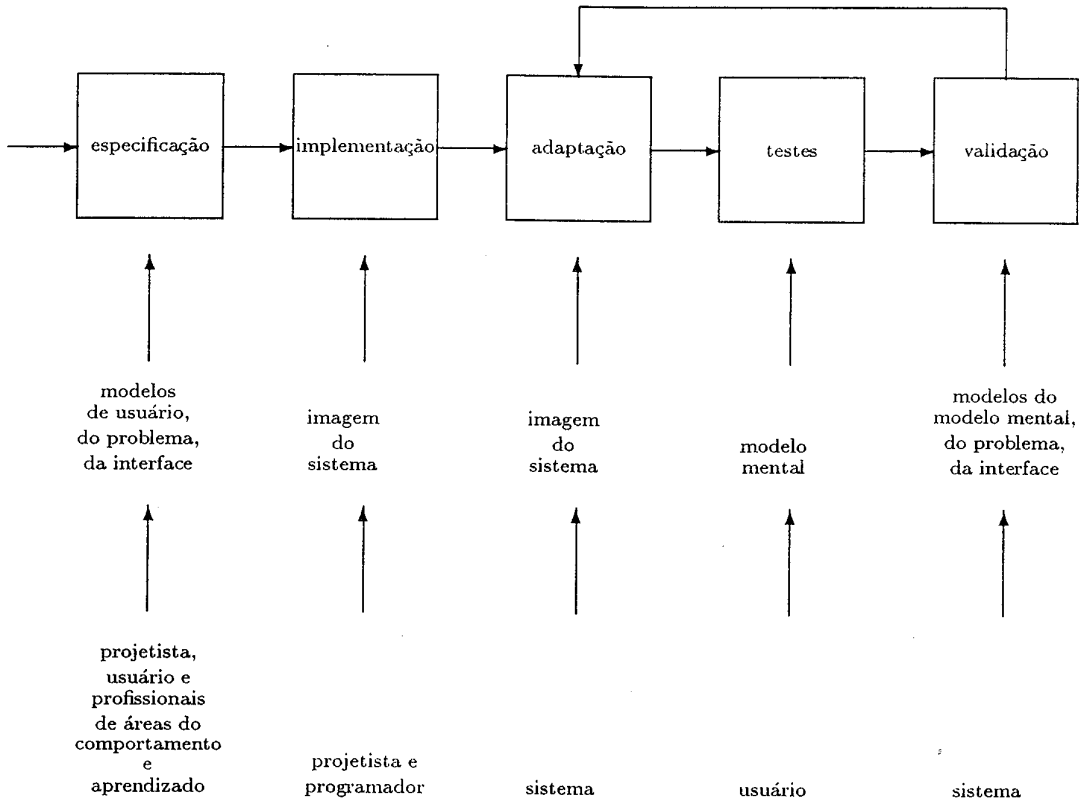
## 2.4 Análise crítica

A modelagem de usuário permite o conhecimento das expectativas, necessidades e dificuldades do usuário em relação ao sistema. Os modelos de usuário, do problema e da interface servem de base inicial para a construção da interface. O modelo mental que o usuário adquire sobre o sistema é fruto da imagem do sistema por ele percebida, ou seja, da interface e do material de instrução para seu uso. Através da análise de experimentos e observações feitos sobre o usuário durante a utilização da interface, e com o auxílio de teorias de cognição, de psicologia do comportamento e de aprendizado, é formado um modelo do modelo mental. Este último modelo permite a reformulação da imagem do sistema adequando-a às necessidades do usuário.

Associado ao ciclo de modelagens de usuário, do problema e da própria interface, está o processo iterativo de especificação, implementação, adaptação, testes e validação da interface. A Figura 2.2 esquematiza este processo iterativo de geração da interface, relaciona cada uma de suas etapas com os responsáveis por sua efetivação, e estes com os modelos por eles criados.

A interface ideal é alcançada quando há consistência entre o modelo de usuário, a imagem do sistema, o modelo mental e o modelo do modelo mental. A consistência entre a imagem do sistema e o modelo de usuário ou o modelo do modelo mental depende do grau de conhecimento que os projetistas e demais profissionais relacionados têm sobre o usuário, da expressividade dos modelos do problema e da interface, e do cuidado com que a interface e o material de instrução para seu uso são construídos.

A consistência entre o modelo mental e a imagem do sistema é obtida se esta imagem for de fácil aprendizado, funcional, útil e confiável, de modo a auxiliar o usuário a desenvolver um modelo mental coerente e correto. Para auxiliar na obtenção desta consistência, além dos cuidados naturais com a modelagem do problema e da interface, é preciso muito cuidado nas decisões de adaptabilidade e partilhamento de responsabilidades conforme será discutido no decorrer deste texto.



*Figura 2.2 Etapas na geração e adequação de interfaces*

A consistência entre o modelo do modelo mental e o modelo mental é alcançada através de experimentos e observações feitos sobre o usuário durante a utilização do sistema, amparados por teorias de cognição, de psicologia do comportamento e de aprendizado. Por sua própria natureza, esta consistência é a mais difícil de ser obtida.

[RICH83] descreve modelagem de usuários em termos de três dimensões, cada uma delas vista sob duas perspectivas. A primeira dimensão focaliza o número de modelos considerados pela interface. Nela é questionada a consideração de um único modelo padrão em contraposição à de vários modelos de usuários individuais. A segunda dimensão focaliza o especificador do modelo. Ela investiga a especificação feita pelo projetista e pelo usuário em contraposição à especificação inferida da observação do comportamento do usuário durante o uso de um protótipo da interface. A terceira dimensão focaliza as características individuais do usuário sob os aspectos de características de longo prazo, como por exemplo, áreas de interesse e grau de experiência, em contraposição às características de curto prazo, como por exemplo, a tarefa que o usuário está tentando resolver.

Este enfoque é interessante por ser aplicável também na discussão das modelagens do problema e da interface. As dimensões citadas podem ser vistas, respectivamente, como modelagem do canônico em contraposição ao particular, modelagem explícita em contra-

posição à implícita, e modelagem de características de longo-prazo em contraposição à de características de curto-prazo. O importante em cada uma delas é a explicitação do modo pelo qual os modelos influenciam a performance do sistema como um todo.

Sistemas que usam um único modelo canônico pecam por falta de adaptabilidade. Sistemas que extraem os modelos em tempo de execução precisam estar aptos a lidar com aspectos incorretos ou informações conflitantes oriundos das inferências que geram os modelos. Sistemas que lidam apenas com características de curto-prazo precisam ser capazes de se adaptar quando estas características são alteradas.

O problema de definir e atualizar os modelos é um problema de projeto, representação e armazenamento de conhecimento e não será tratado neste texto.

### 3 Tipos de diálogo

Num modelo de interação pessoa-computador existem os agentes usuário, diálogo e processador de tarefas [CHIG89], [CARD89]. O diálogo é responsável pela comunicação entre o processador de tarefas e o usuário para possibilitar o desempenho das funções pertinentes a cada um. Ele é caracterizado pelos aspectos funcional e linguístico.

#### 3.1 O aspecto funcional

O aspecto funcional é definido pelos objetivos a serem alcançados através do diálogo. Naturalmente, um objetivo que deve estar por trás de qualquer diálogo é auxiliar o usuário no processo de comunicação com o processador de tarefas. Este auxílio pode ser proporcionado com o objetivo de otimizar a performance do sistema (usuário e processador de tarefas), ou de alterar o processo cognitivo do usuário. No primeiro grupo enquadram-se os sistemas denominados assistentes, entre os quais estão, por exemplo, os sistemas de suporte para atividades criativas, os sistemas de auxílio à decisão e os sistemas especialistas que atuam como um assistente técnico [COOM84], [WATE86], [ROTH87], [KAIS88]. No segundo grupo enquadram-se os sistemas tutoriais que ensinam ao usuário o uso específico da interface [CARR84], [DIGI86], ou que atuam como um professor de áreas específicas [WENG87], como por exemplo, medicina, matemática ou química.

Tanto o assistente quanto o tutor podem atuar de modo passivo ou ativo [CARD89], [ELKE89]. No modo passivo, o diálogo permite que a comunicação ocorra sob completo e contínuo controle do usuário. No modo ativo, o diálogo pode controlar completamente a comunicação, fazendo com que o usuário atue apenas como um fornecedor de dados, ou pode controlá-la parcialmente. Quaisquer destes modos podem ocorrer através de diferentes níveis de automatização [CARD89]. Por exemplo, um diálogo passivo pode oferecer ou não algum método de auxílio para que o usuário selecione as ações que deve tomar; um diálogo que controla completamente a comunicação pode ser obrigado a informar ao usuário o que foi feito, ou ter o poder de decidir se informa ou não; um diálogo que controla parcialmente a comunicação pode selecionar uma ação e consultar o usuário antes de implementá-la, ou implementá-la diretamente mas permitir que o usuário a interrompa.

## 3.2 O aspecto linguístico

Sob o aspecto linguístico, o diálogo é composto pelas linguagens de discurso de usuário e de discurso de processador de tarefas, e se ocupa em traduzir uma linguagem para a outra. A linguagem de discurso de usuário é a linguagem de interação vista e entendida pelo usuário; a linguagem de discurso de processador de tarefas é a linguagem de interação vista e entendida pelo processador de tarefas. A tradução de uma linguagem para outra corresponde a um mapeamento entre as funções do processador de tarefas e os comandos disponíveis para o usuário. A Figura 3.1 representa este modelo.

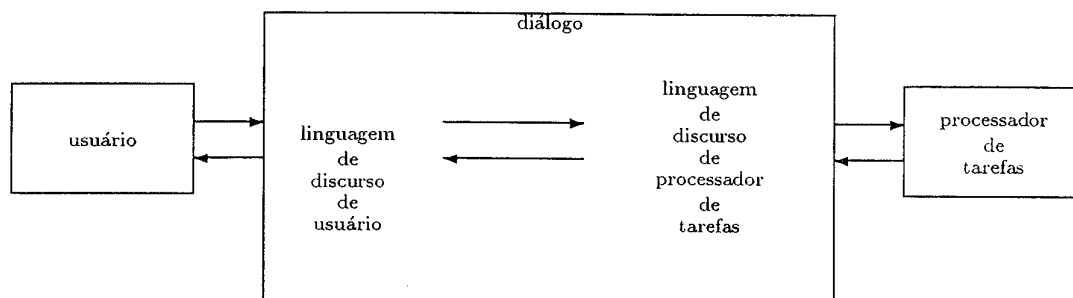


Figura 3.1 Modelo de interação pessoa-computador

A linguagem de discurso do processador de tarefas é totalmente dependente da tarefa em questão e não será focalizada neste texto. A linguagem de discurso de usuário, também conhecida como estilo de interação [SHNE87], [BAECS7], é caracterizada pela forma da linguagem através da qual o usuário manda executar as ações desejadas e recebe informação sobre os resultados destas ordens, e pela forma de acesso aos dados e objetos do processador de tarefas. Cada estilo de interação tem características fundamentais como grau de flexibilidade, quantidade de itens que o usuário tem de memorizar, visibilidade dos níveis de indireção ou *modos* do processador de tarefas e suporte de *software* e *hardware*. A escolha do estilo de interação é uma tarefa complexa que depende da adequação das características do estilo ao problema alvo do processador de tarefas e aos seus usuários efetivos.

### 3.2.1 A forma da linguagem

Quanto à forma da linguagem, os estilos de interação existentes podem ser classificados em linguagens de comandos e de seleção em *menus*. Alguns aspectos básicos devem ser considerados independentemente do estilo escolhido, para que estas linguagens sejam úteis e permitam a comunicação efetiva entre homem e computador. Entre estes aspectos citam-se os correspondentes às *máximas* que devem ser atendidas em qualquer diálogo governado pelo Princípio da Cooperação, conforme [GRIC75]. Atendendo às *máximas*, a forma da linguagem deve ser:



1. compacta, no sentido de ser tão informativa quanto seja requerido e não mais que isto;
2. significativa, no sentido de ter sua estrutura organizada semanticamente, ser relevante no contexto em que se insere e apresentar apenas o que pode ser usado;
3. precisa, no sentido de não ser obscura ou ambígua.

Como toda linguagem, a linguagem de discurso de usuário tem estruturas léxica, sintática e semântica que precisam ser conhecidas e memorizadas por seus usuários. No nível semântico, é definido um modelo de objetos e ações, que deve ser o mais próximo possível da realidade. No nível léxico, devem ser escolhidos nomes distintos, específicos e significativos ao usuário, correspondentes ao modelo de objetos e ações já definidos. No nível sintático, é definida uma estrutura hierárquica e consistente, através da hierarquização dos comandos, ordenação de seus argumentos e consistência com as relações ação-objeto.

O número de comandos de uma linguagem depende do número de funções do processador de tarefas. Para manter pequeno o número de comandos da linguagem, pode ser interessante dar diferentes interpretações para um mesmo comando conforme o contexto em que ele seja acionado. Esta variação de interpretação de um comando corresponde ao conceito de *modo* do processador de tarefas. Se esta variação de *modo* for necessária, ela deve ocorrer da forma mais visível e clara possível para que o usuário não a perceba como uma barreira.

**Linguagens de comandos:** Em um diálogo com o estilo de linguagem de comandos, o usuário digita os comandos desejados em uma linguagem verbal previamente definida. As linguagens de comandos podem ser divididas em linguagens artificiais e linguagens pseudo-naturais (imprópria e usualmente denominadas linguagens naturais).

Linguagens artificiais são próximas às linguagens de programação. Um diálogo com este estilo de interação deve [VASS84], [SHNE87]:

- dar suporte ao usuário para uso de regras de abreviação dos comandos;
- oferecer ao usuário a capacidade de criar macros para sequências de comandos;
- ser de fácil escrita e leitura.

Linguagens naturais correspondem a subconjuntos significativos e bem definidos de uma linguagem natural, como por exemplo do inglês ou português. [RICH84] descreve algumas condições sob as quais linguagem natural pode ser efetiva como um meio de entrada para o computador. Um diálogo em linguagem natural deve ser capaz de:

- interpretar corretamente as frases que expressam os comandos para permitir comunicação robusta;
- explicar suas ações e respostas;

- manter coerência durante a comunicação;
- identificar objetos a partir das suas descrições;
- gerar descrições apropriadas ao contexto.

A interação em linguagem natural apresenta dificuldades sérias que não ocorrem normalmente em linguagens artificiais [RICH84], [ADAR84], [LITMaa], [JAME86], [KENN88]. Entre estas dificuldades destacam-se a complexidade semântica e a falta de concisão e de precisão inerentes a linguagens naturais.

Na interação em linguagem natural há necessidade de limitar o grau de cobertura da linguagem origem, exigindo que o usuário aprenda qual é o subconjunto válido para a interação. Portanto, seja em linguagem artificial ou seja em linguagem natural o usuário precisa aprender uma linguagem nova.

**Seleção em menus:** Nesta forma de linguagem, o usuário seleciona o comando desejado em um *menu* que lhe é apresentado na tela. Cada item de *menu* corresponde a um comando disponível no processador de tarefas. Um diálogo neste estilo deve:

- ter tempo de resposta e taxa de exibição rápidos;
- ter a estrutura de *menus* organizada semanticamente e de modo visível para o usuário (por exemplo, ítems serem títulos quando caminhando na estrutura de *menus*);
- ter cada *menu* com ítems agrupados de modo significativo;
- proporcionar formas de abreviar a entrada a *menus* muito frequentes (por exemplo, permitindo o retorno ao *menu* imediatamente anterior e ao principal);
- apresentar *layout* e terminologia consistentes entre si e apropriados para a tela;
- oferecer facilidades para auxiliar o usuário a entender a função de cada item e a reparar seus erros.

Os *menus* podem ter formato textual ou icônico, conforme seus ítems sejam palavras e frases ou símbolos gráficos, respectivamente. Existe pouca documentação que evidencie as vantagens e desvantagens entre estes dois formatos de *menus*. Entre os argumentos a favor de ícones [LODD83], [GIT86] destacam-se a maior universalidade de ícones em relação à de textos e o poder de memorização pictórica das pessoas. Entre os argumentos contra o uso de ícones [MANE85], [GIT86] destacam-se a ineficiência para lidar com grande número de comandos similares, o risco de o usuário inferir incorretamente o significado do ícone e o maior consumo de espaço na tela. Para reduzir o risco de interpretação incorreta do ícone, devem ser tomadas as seguintes regras como básicas [POTO88]:

- fazer ícones fáceis de entender;

- não usar analogias confusas;
- não violar estereótipos populares.

Os diálogos com linguagens de seleção em *menus* apresentam algumas vantagens e outras desvantagens em relação aos diálogos com linguagens de comandos. Entre as vantagens destacam-se a pequena quantidade de comandos que o usuário precisa memorizar, a possibilidade de o usuário aprender rapidamente a utilizar a linguagem, o número reduzido de digitação de teclas e, conseqüentemente, a redução do número de erros cometidos pelo usuário.

Entre as desvantagens, destacam-se o maior consumo de espaço na tela, a exigência de mais rápida taxa de exibição e a menor flexibilidade para a entrada de comandos, impossibilitando o usuário de definir macros que possam acelerar o seu trabalho.

### 3.2.2 O acesso a dados e a objetos

O acesso aos dados e aos objetos do processador interativo de tarefas pode ser percebido pelo usuário como indireto ou direto. O acesso indireto é a forma tradicional em que o usuário manda que o processador de tarefas execute ações sobre dados nem sempre visíveis ao usuário. O acesso direto [SHNE83] corresponde à técnica denominada manipulação direta, na qual o usuário opera diretamente sobre os dados ou objetos exibidos na tela.

Um diálogo com manipulação direta deve:

- apresentar boas metáforas para representar o mundo da aplicação em termos de objetos na tela e de ações de entrada;
- representar continuamente o objeto de interesse;
- permitir operações rápidas, incrementais e reversíveis cujo impacto sobre o objeto de interesse seja imediatamente visível;
- assumir a complementaridade entre entrada e saída, isto é, os resultados de uma manipulação exibidos na tela são aceitáveis como entrada para manipulações seguintes.

Quaisquer das formas de linguagem descritas no Item 3.2.1 podem ser adotadas em ambas as formas de acesso aos dados. Entretanto, os diálogos com manipulação direta têm adotado mais frequentemente seleção em *menus*. Tipicamente, um diálogo com manipulação direta contém um conjunto de objetos e *menus* de manipulações que podem ser feitas sobre eles, ambos exibidos simultaneamente na tela [SHNE87], [JACO89]. É claro que se a linguagem do processador de tarefas tiver grande número de comandos, nem todos eles caberão na tela e haverá necessidade de uma hierarquização, com introdução de modos, estados, ou sub-diálogos.

As principais vantagens do acesso direto aos objetos em relação ao acesso indireto são a habilidade de manipular diretamente os objetos exibidos e a transparência das representações destes objetos e das manipulações sobre eles. Estas duas vantagens têm um

crédito psicológico e cognitivo positivo por parte do usuário, pois lhe proporcionam uma sensação de segurança e reduzem o seu esforço mental em traduzir as ações de entrada e representações de saída para as operações e objetos no domínio do problema.

### 3.3 Análise crítica

A definição de um tipo de diálogo se dá pela combinação de uma característica funcional com uma linguística, de modo a atender às necessidades do usuário e ser natural ao domínio do problema. Esta adequação é complexa pois envolve variáveis nem sempre controláveis, como por exemplo, a personalidade do usuário e situações imprevistas que podem ocorrer durante o uso do sistema.

Em [ROTH87], por exemplo, é relatada uma experiência na qual técnicos de diferentes personalidades e níveis de experiência usaram um sistema especialista como um assistente técnico para diagnosticar falhas em um determinado equipamento. O sistema havia sido desenvolvido sob um paradigma no qual o usuário atua como um simples fornecedor de dados, deixando que o sistema se encarregue de toda a tarefa de solução do problema. Os resultados da experiência permitem concluir que:

- é impossível o sistema prever e tentar controlar todas as possíveis situações em um domínio de problema. Portanto, o sistema deve ser capaz de partilhar o controle com o usuário, permitindo que ele tenha a responsabilidade de detectar e sair de caminhos improdutivos porventura tomados pelo sistema.
- a atitude ativa do usuário pode ser mais significativa que seu nível de experiência para obtenção de sucesso nas interações.

Não obstante, a contraposição das características dos tipos de usuários definidas na Seção 2 e das características dos tipos de diálogo aqui discutidas permite fazer algumas observações que podem auxiliar na adequação entre tipo de diálogo e de usuário. Quanto à funcionalidade do diálogo, pode-se ter como princípio que:

- sistemas tutoriais são adequados a usuários inexperientes nos temas específicos por eles tratados;
- sistemas assistentes podem ser adequados para usuários com qualquer nível de experiência;
- sistemas que controlam parcialmente a comunicação podem ser adequados para quaisquer tipos de usuários.

Quanto ao aspecto linguístico, pode-se dizer que, em princípio, o acesso a dados e a objetos via manipulação direta é interessante e atraente para qualquer tipo de usuário. A forma da linguagem no estilo de interação tem variações maiores que devem ser consideradas conforme o tipo de usuário.

A interação via linguagens de comandos artificiais possibilita flexibilidade e uma grande sensação de controle e iniciativa para os usuários experientes. Entretanto, para usuários

novatos ou ocasionais este estilo pode causar ansiedade e insegurança devido à necessidade de memorizar comandos.

É discutível o interesse da interação via linguagens de comandos naturais para os usuários novato e ocasional. Por um lado, ela parece uma boa opção por ser muito explicativa e por sua aparência natural. Por outro lado, ela também exige a memorização do subconjunto da linguagem origem que é aceito no diálogo, o que pode ocasionar insegurança e decepções ao usuário. Para o usuário experiente, a inadequação pode se verificar pela falta de concisão e de flexibilidade para acelerar suas tarefas.

Independentemente do nível de experiência do usuário, o sucesso da interação via linguagem natural depende também da confiança que ele tenha na capacidade de comunicação do sistema. Em [KENN88], por exemplo, são relatadas experiências de diálogos em linguagem natural, nas quais os usuários usavam o computador como meio de comunicação [KAMM88] e acreditavam que seu parceiro era um sistema de computação ou um outro usuário. As análises foram concentradas no uso de anáforas e na escolha das palavras. Embora em todas as experiências o parceiro do usuário fosse realmente uma outra pessoa, o comportamento dos usuários teve variações dependentes de com quem eles acreditavam estar conversando. Quando o usuário acreditava que seu parceiro era um sistema de computação ele, invariavelmente, assumia que o sistema tinha apenas um conhecimento restrito sobre o conteúdo do diálogo, e tinha meios limitados para compreendê-lo e representá-lo. Conseqüentemente, o usuário fazia frases curtas, restringia o número de palavras a um mínimo *negociado*, e minimizava o uso de anáforas pronominais. Apesar de os diálogos resultantes serem menos elegantes que aqueles entre pessoas, não se pode concluir que eles eram menos informativos.

A interação via seleção de *menus* é interessante para usuários novatos e ocasionais por exigir pouca memorização e pouco treinamento. Se os mecanismos de exibição e seleção forem muito rápidos, esta forma de linguagem pode ser interessante também para usuários experientes.

Combinado com qualquer das formas de linguagem acima, para facilitar a entrada de dados, pode ser interessante a utilização de preenchimento de campos de formulários [SHNE87] exibidos na tela. Naturalmente, o sucesso deste auxílio depende da organização semântica do formulário, da expressividade dos nomes dos seus campos e da clareza das instruções de preenchimento.

Sistemas de computação interativos usam frequentemente ambientes de janelas nos quais a tela do usuário é dividida em áreas retangulares que podem se sobrepor e suportar diálogos diferentes [CARD84], [GOLD84]. Entre as motivações para uso de janelas está a possibilidade de evidenciar as mudanças de modo no processador de tarefas e a sua organização semântica.

Todos os estilos de interação aqui considerados supõem a utilização da tela como dispositivo de saída e do teclado, *mouse* ou algum outro dispositivo de apontamento como dispositivo de entrada. A evolução tecnológica deverá enriquecer o diálogo entre usuário e computador pela combinação dos estilos de interação aqui discutidos com a sintetização de som e o reconhecimento de voz [SIMP85].

Para que a interface possa atender a vários tipos de usuário, é interessante que ela disponha

simultaneamente de vários tipos de diálogo. Além das considerações anteriores, na escolha dos estilos de interação, devem ser considerados ainda:

- o domínio do problema – o estilo deve ser natural ao domínio do problema. Por exemplo, não faz sentido usar ícones para expressar operações aritméticas.
- o custo de implementação do estilo – entre os estilos discutidos, os mais caros são os que usam linguagem natural ou manipulação direta. Isto ocorre principalmente porque existe pouco ou nenhum *software* básico para auxiliar na programação destes estilos, diferentemente do que ocorre para os demais.
- as características do *hardware* disponível – diálogos icônicos requerem tela gráfica e de alta resolução.

Certamente diálogos, quando adequados ao tipo de usuário, facilitam a comunicação pessoa-computador e promovem prazer na experiência de sua utilização. Entretanto, o verdadeiro papel dos diálogos é o de serem os veículos para conduzir os dados ou o conhecimento.

## 4 Gerência da interface

Para que a interface mantenha o curso do diálogo e contribua relevantemente na interação, ela precisa ser capaz de reconhecer as intenções de comunicação do usuário [NORM84], [BACHaa] no contexto da interação [CROF84]. Este reconhecimento pode ser complexo pois estas intenções geralmente são acompanhadas por intenções de prosseguimento no diálogo e fazem parte de planos maiores.

O contexto da interação é definido pelas características do ambiente que são importantes na determinação do fluxo da interação. Entre estas características estão uma identificação do estado corrente do usuário e da tarefa em execução.

A identificação do estado corrente do usuário pode ser obtida por um questionamento direto ao usuário ou por inferências a partir de *feedback* dado por ele. Em ambos os casos pode ser considerada a modelagem de usuário discutida na Seção 2. No primeiro caso, a identificação corresponde a um estereótipo definido no modelo de usuário; no segundo, ela corresponde ao modelo do modelo mental que atualiza o modelo de usuário pela análise do histórico da interação. Esta identificação é importante para auxiliar na adaptação do tipo de diálogo conforme as preferências individuais do usuário.

Supondo que a interface tenha acesso a vários tipos de diálogo e conheça a classificação do usuário, podem ser destacadas três abordagens de adaptação. Na primeira e mais simples abordagem é usado um modelo normativo [CARR84] que delimita as funções disponíveis para cada tipo de usuário. Entretanto, esta abordagem pode frustrar o usuário, por exemplo, não permitindo que um novato acesse uma função avançada específica.

Na segunda abordagem, o conhecimento do novato é visto como um subconjunto do conhecimento do experiente [CARR87], e a transição de novato para experiente se dá

por acréscimo de conhecimento, minimizando gradualmente as diferenças de desempenho entre os dois tipos de usuário. Entretanto, teorias de psicologia do aprendizado, tais como estímulo-resposta, indicam que o aprendizado se faz por reestruturação de conhecimento e não apenas por acréscimo.

Na terceira abordagem é feita uma atribuição de pesos às funções do sistema, de modo que as de maior peso possam ser alteradas apenas com consentimento do usuário. Para auxiliar na definição de pesos das funções, pode ser interessante a identificação de um homomorfismo [MORA87] natural ao sistema. As funções presentes no homomorfismo de mais alto nível são aquelas em que o usuário precisa confiar e devem ter maior peso. A identificação do homomorfismo pode ocorrer na fase de projeto para obtenção de um protótipo. Este protótipo pode ser alterado em tempo de execução pela análise do histórico da interação feita, por exemplo, com base na frequência relativa das funções que o usuário utiliza.

A identificação da tarefa em execução pode ser obtida por um mecanismo de inferência sobre as modelagens do problema e do processador de tarefas, em conjunto com o histórico da interação corrente. Um mecanismo de inferência que tem sido explorado em programas de inteligência artificial é o reconhecimento de planos de domínio [RICH88], [BACHaa], [LITMaa], [JAME86]. Estes planos são estruturas que representam tarefas típicas de um dado domínio e correspondem aos tópicos de diálogos sobre aquele domínio. Eles são sequências de ações e estados conectados por relações de causa. Usualmente, modelos de planos contêm pré-requisitos, efeitos e uma decomposição em ações primitivas ou outros planos. Pré-requisitos são condições que precisam ocorrer antes que o plano possa ser aplicado; efeitos são comandos do processador de tarefas que serão executados depois que o plano tenha sido aplicado com sucesso. A tarefa do reconhecedor de planos é reconhecer uma estrutura do plano do domínio à qual as intenções do usuário se referem.

**Análise crítica:** A adequação da interface às necessidades do usuário engloba uma adaptação do estilo de diálogo e um partilhamento de controle da interação, de modo sensível ao contexto.

Nas interfaces tradicionais, as decisões sobre a adequação são tomadas na fase de projeto, gerando interfaces flexíveis [ELKE89]. Nestas interfaces, diferentes métodos de interação podem ser usados pelo usuário para executar uma tarefa conforme a identificação de seu nível de experiência e de suas preferências. Entretanto, esta flexibilidade não é suficiente devido à evolução dos usuários durante a interação e à possibilidade de ocorrerem situações imprevistas. Isto sugere que as decisões sobre a adequação sejam tomadas dinamicamente pela interface, possivelmente, em conjunto com o usuário, conforme o contexto da interação.

A adequação pode ser feita em diferentes níveis de complexidade, conforme a estrutura do tipo de diálogo usado. Por exemplo, diálogos no estilo de *menus* de opções podem ser estruturados de forma hierárquica, possivelmente correspondendo à hierarquia de planos de domínio. Por um lado, esta estrutura facilita o reconhecimento das intenções do usuário durante uma interação. Por outro lado, ela pode enrijecer o diálogo e dificultar o tratamento de situações imprevistas.

Em diálogos em linguagem natural, a dificuldade maior reside no reconhecimento das intenções de comunicação do usuário [BACHaa], [LITMaa], [JAME86], pois elas não ne-

cessariamente são explícitas e nem precisam ter uma estrutura hierárquica coincidente com planos do domínio ou da tarefa. Não obstante, a abordagem de reconhecimento de planos tem sido explorada para auxiliar no reconhecimento das intenções de comunicação.

Em [LITMaa], por exemplo, é apresentado um modelo de reconhecimento de planos para entendimento de diálogos orientados para tarefas que separa explicitamente o que é dito sobre a tarefa e o que é consensualmente conhecido sobre ela. Esta separação é possibilitada pela definição de um conjunto de planos, planos de discurso, que podem ser gerados a partir da execução ou discussão de planos de domínio, mas de modo que ambos os conjuntos de planos sejam distintos um do outro. Além disto, as intenções de discurso são distinguidas dos planos aos quais elas se referem. A incorporação do conhecimento sobre os planos de discurso permite a consideração de sub-diálogos, tais como explicações e correções, e a manutenção das vantagens computacionais da abordagem baseada em planos.

## 5 Uma arquitetura

A interface é inteligente se ela, contínua e permanentemente, é capaz de adquirir, aplicar e propagar conhecimentos [OGBO83]. Para isto, ela precisa ter meios de representar conhecimentos, meios para raciocinar sobre estes conhecimentos e meios para se expressar e adquirir novos conhecimentos. Isto induz à definição da arquitetura mínima para interfaces inteligentes mostrada na Figura 5.1.

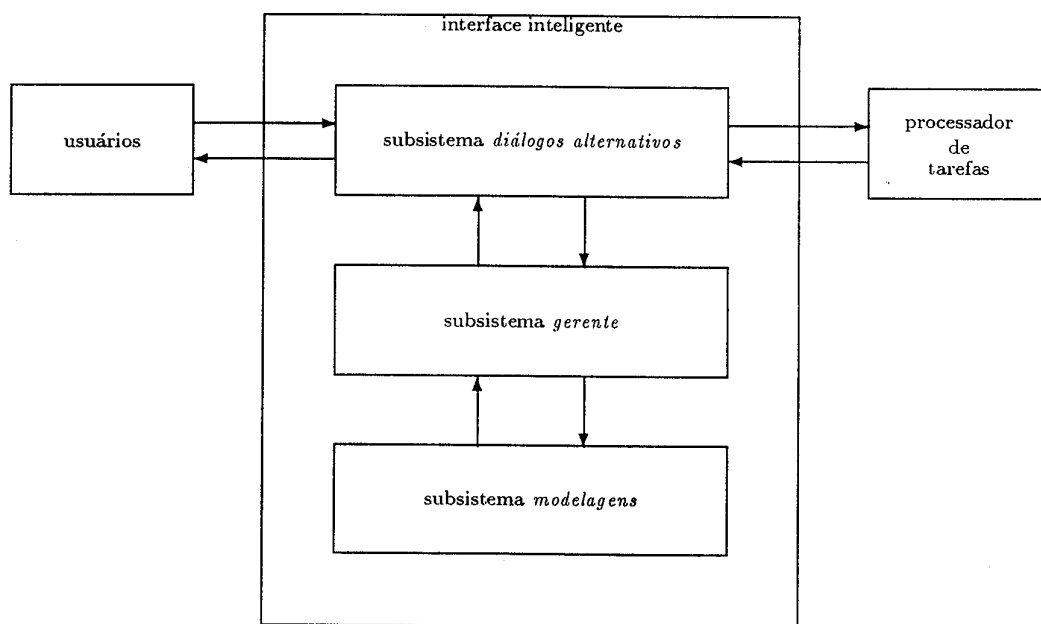


Figura 5.1 Uma arquitetura para interfaces inteligentes



O subsistema *modelagens* constrói modelos de representação de conhecimentos sobre o usuário, sobre o domínio do problema e sobre a interface. Estes modelos são básicos para a adequação da interface às necessidades do usuário e devem estar sujeitos a contínuas modificações resultantes de *feedback* nas interações.

O subsistema *diálogos alternativos* é responsável pela efetivação e visualização da troca de símbolos entre o usuário e o processador de tarefas. Nele estão encapsulados os vários tipos de diálogo disponíveis na interface e que lhe permitem se expressar e adquirir novos conhecimentos.

O subsistema *gerente* administra os subsistemas *modelagens* e *diálogos alternativos* de modo a permitir que o diálogo flua de modo natural e cooperativo. Para isto, ele precisa possuir um mecanismo de inferência que lhe permita adequar o tipo de diálogo ao usuário, dinamicamente, conforme o contexto da interação. Além disto, ele precisa de mecanismos de filtragem que lhe permitam obter no decorrer do diálogo as informações relevantes para a realimentação do subsistema de modelagens. Ou seja, ele é o meio pelo qual a interface é capaz de raciocinar sobre as fontes de conhecimento às quais tem acesso.

**Análise crítica:** A arquitetura definida apresenta aspectos críticos, tais como, risco de conflitos pela modificabilidade permanente dos modelos presentes no subsistema *modelagens*, sobrecarga do sistema pelo excesso de tipos de diálogo, dificuldades de partilhamento de controle em tempo de execução, dificuldade de a interface se auto-reproduzir [OGBO83].

Para reduzir o risco de conflitos gerados pelas modificações nos modelos do subsistema *modelagens*, a interface pode associar um peso de credibilidade a cada um dos aspectos considerados em seus modelos, de modo a distinguir os conhecimentos inferidos dos tidos como verdade absoluta.

A sobrecarga do sistema pelo excesso de tipos de diálogo pode ser reduzida pelo uso do paradigma de orientação a objetos [GOLD84], [COXB86], [SZEK89], [TYLE89]. Este paradigma permite a otimização da utilização de código, por exemplo, através do mecanismo de polimorfismo para minimizar o número de mensagens necessárias no protocolo de comunicação para traduzir a linguagem de discurso de usuário em linguagem de discurso do processador de tarefas.

Decisões sobre partilhamento de responsabilidades são difíceis se tomadas em tempo de especificação do sistema devido à necessidade de atender às expectativas do usuário e ao fato de estas dependerem da sua personalidade. Mais difícil ainda é alterar o partilhamento de responsabilidades em tempo de execução do sistema. Neste caso, além da dificuldade anterior, há o acréscimo do aspecto adaptação à evolução do usuário. Decisões sobre como e quando se adaptar ao usuário são críticas por estarem diretamente relacionadas à confiança [MUIR87] do usuário no sistema. Esta confiança depende da percepção que o usuário tem sobre a previsibilidade do comportamento do sistema.

Como a confiança no sistema é essencial para seu sucesso, é fundamental que a adaptação ao usuário seja feita de modo por ele previsível. Para isto, ou o sistema informa ao usuário as alterações específicas que vão ocorrer, ou a adaptação deve ser feita apenas em funções do sistema que não coloquem em risco a confiança do usuário.

A capacidade de, contínua e permanentemente, adquirir, aplicar e propagar conhecimentos pode ser definida como capacidade de auto-reprodução [OGBO83]. Para se auto-reproduzir a interface precisa ter pelo menos dois níveis de processos: um nível para lidar diretamente com o usuário e um nível para, a partir do *feedback* dado pelo usuário, ela poder construir novos processos.

A definição desta arquitetura é interessante para direcionar pesquisas futuras para a obtenção de interfaces inteligentes [ELKE89]. Por exemplo, ela evidencia a necessidade de pesquisas sobre:

- mecanismos de representação de conhecimento;
- mecanismos de inferência para aquisição e combinação de conhecimentos;
- mecanismos para permitir auto-reprodução da interface;
- mecanismos para permitir comunicação robusta com o usuário.

## 6 Conclusões

A Figura 6.1 esquematiza a visão de interfaces inteligentes dada neste texto. Uma interface inteligente deve atuar entre o usuário e o processador de tarefas de modo a se adaptar, dinamicamente, às necessidades do usuário conforme o contexto da interação. Para isto, ela precisa de um *mecanismo de expressão*, de uma *base de conhecimento* e de um *mecanismo de inferência*. Estes componentes correspondem aos subsistemas *diálogos alternativos*, *modelagens* e *gerente* discutidos na Seção 5.

O subsistema *diálogos alternativos* deve conter os tipos de diálogo pertinentes ao domínio do problema e disponíveis na interface. Um tipo de diálogo é composto por um aspecto linguístico e um funcional. O aspecto linguístico permite a veiculação dos dados entre o usuário e o processador de tarefas, preferivelmente de modo agradável ao usuário. O aspecto funcional possibilita que esta veiculação dos dados ocorra de modo útil para promover o desempenho do sistema (usuário, interface, processador de tarefas).

O subsistema *modelagens* deve conter modelos do usuário, do problema e da interface. Cada um destes modelos deve ter uma parte fixa que é pré-definida e assumida como verdade absoluta e uma parte mutável para possibilitar o armazenamento dos conhecimentos adquiridos pelo uso da interface.

O subsistema *gerente* deve conter heurísticas e regras de inferência que lhe permitam tomar decisões sobre a adaptação da interface ao usuário e filtrar informações para realimentar o subsistema *modelagens*. A adaptação pode ser feita no nível de tipo de diálogo e no nível de partilhamento de controle da interação. Em ambos os casos, devem ser consideradas as características do usuário e da tarefa em execução, juntamente com as capacidades do processador de tarefas e da interface.

As técnicas de adaptação discutidas na Seção 4 são usadas normalmente na fase de projeto e possibilitam gerar interfaces flexíveis. Se usadas em tempo de execução do sistema, e

em conjunto com heurísticas que raciocinem sobre os modelos e o histórico da interação, estas técnicas podem ser úteis para gerar interfaces inteligentes.

Além da análise crítica da literatura, este texto pretende ter contribuído para o estabelecimento de uma taxonomia de tipos de diálogo, e para a identificação de uma arquitetura mínima necessária à implementação de interfaces inteligentes.

Em suma, o projeto de interfaces inteligentes, conforme a conceituação aqui discutida, é complexo e inerentemente multidisciplinar, exigindo a integração e colaboração de profissionais das áreas de inteligência artificial, engenharia de *software*, *design*, psicologia, ciência cognitiva, ergonomia, linguística e *hardware*, além da área específica do problema do programa de aplicação.

A principal dificuldade na geração de interfaces inteligentes está na compreensão, por parte da interface, do nível conceitual do diálogo. Esta compreensão é possível à medida em que a interface é capaz de adquirir conhecimentos e de raciocinar sobre eles durante a interação e isto, na arquitetura proposta, é desempenhado pelos subsistemas *modelagens* e *gerente*. O projeto destes subsistemas pode ser auxiliado por modelos computacionais definidos com base no processo cognitivo pelo qual passa o ser humano [NADI87], [TAKA87], [BENT89], e é objeto de outro trabalho, ainda em desenvolvimento.

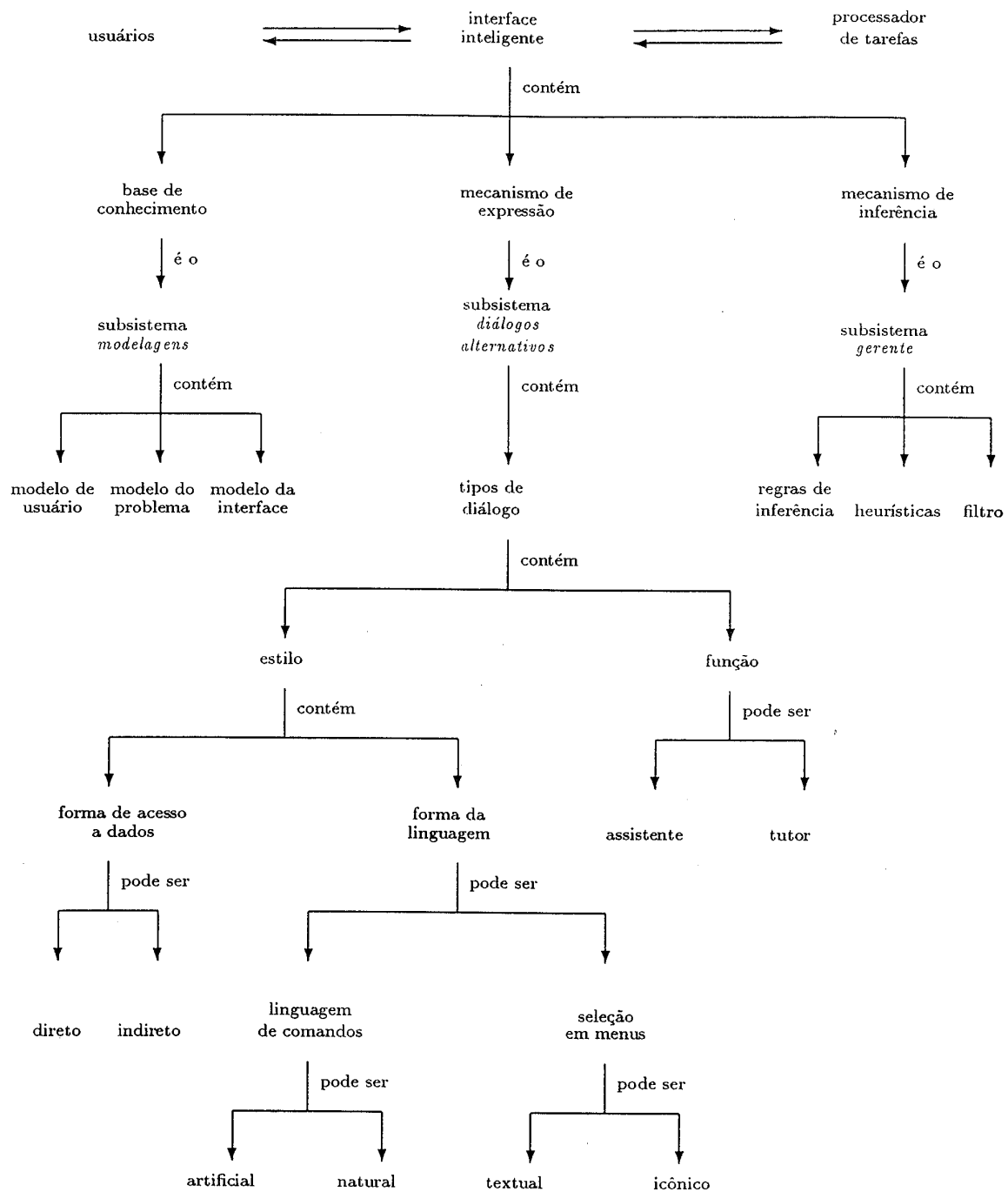


Figura 6.1 Interfaces inteligentes

## Referências

- [ADAR84] Reichman-Adar, R. Extended Person-Machine Interface. *Artificial Intelligence* 22, 1984, 157-218.
- [AIR85a] Airchinnigh, M.M.; Report on the User's Conceptual Model. In Pfaff, G.E. (Ed.) User Interface Management Systems, *Proceedings of the Workshop on User Interface Management Systems*, Seeheim, FRG, November 1-3, 1983; Eurographics, 1985, 31-40.
- [AIR85b] Airchinnigh, M.M.; A Model of User's Conceptual Model of... . In Pfaff, G.E. (Ed.) User Interface Management Systems, *Proceedings of the Workshop on User Interface Management Systems*, Seeheim, FRG, November 1-3, 1983; Eurographics, 1985, 203-224.
- [APPE85] Apperley, M. & Spence, R. The role of a User's System Model, and its Relevance to User Interface Management. In Pfaff, G.E. (Ed.) User Interface Management Systems, *Proceedings of the Workshop on User Interface Management Systems*, Seeheim, FRG, November 1-3, 1983; Eurographics, 1985, 195-202.
- [BACHaa] Bach, K. Communicative Intentions, Plan Recognition, and Pragmatics: Comments on Thomason and on Litman and Allen. Fonte e data ilegíveis.
- [BAEC87] Baecker, R.M. & Buxton, W.A.S. (writers and editors) Readings in Human-Computer Interaction: A Multidisciplinary Approach. Morgan Kaufmann Publishers, Inc., 1987.
- [BENT89] Bento, J., Feijó, B. & Dowling, P.J. Knowledge based design of steel portal frames for agricultural buildings. Comunicação pessoal, 1989.
- [BRAJ86] Brajnik, G., Giovanni, G. & Tasso, C. An Expert Interface for Effective Man-Machine Interaction. In Bolc, L. & Jarke, M. (Eds.) Cooperative Interfaces to Information Systems. Springer-Verlag Berlin Heidelberg 1986, 259-308.
- [CARD84] Card, S.K., Pavel, M. & Farrell, J. Window-based computer dialogues, *Proceedings of the INTERACT'84*, 1984, 355-359. Amsterdam, The Netherlands: Elsevier.
- [CARD89] Card, S.K., Human Factors and Artificial Intelligence. In Hancock, P.A. & Chignell, M.H. Eds. Intelligent Interfaces Theory, Research and Design. Elsevier Science Publishers B.V., 1989, 27-48.
- [CARR84] Carrol, J.M. & Carrithers, C. Training wheels in a user interface, *Communications of the ACM*, vol 27, no 8, August 1984, 800-806.
- [CARR87] Carrol, J.M. & McKendree, J. Interface Design Issues for Advice-Giving Expert Systems, *Communications of the ACM*, vol 30, no 1, January 1987, 14-31.

- [CARV88] Carvalho, Eliane B.S. Trabalho final de curso em Tópicos em Inteligência Artificial II, PUC-RJ, 1988.
- [CHIG89] Chignell, M.H., Hancock, P.A. & Loewenthal, A. An Introduction to Intelligent Interfaces. In Hancock, P.A. & Chignell, M.H. Eds. *Intelligent Interfaces Theory, Research and Design*. Elsevier Science Publishers B.V., 1989, 1-26.
- [CONA76] Conant, R.C. Laws of information which govern systems, *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6, 1976, 240-255.
- [COOM84] Coombs, M.J. & Alty, J.L. Expert Systems: An alternate paradigm, *International Journal of Man-Machine Studies*, 20, 1984, 21-43.
- [COXB86] Cox, B.J. *Object-Oriented Programming: An Evolutionary Approach*. Addison Wesley Publishing Company, 1986.
- [CROF84] Croft, W.B. The role of context and adaptation in user interfaces, *International Journal of Man-Machine Studies*, 21, 1984, 283-292.
- [DIGI86] Digitalk Inc. *Smalltalk/V Tutorial and Programming Handbook*, 1986.
- [EBER89] Eberts, Ray E. & Eberts, Cindelyn G. Four approaches to human computer interaction. In Hancock, P.A. & Chignell, M.H. Eds. *Intelligent Interfaces Theory, Research and Design*. Elsevier Science Publishers B.V., 1989, 69-128.
- [ELKE89] Elkerton, J. & Williges, R.C. Dialogue Design For Intelligent Interfaces. In Hancock, P.A. & Chignell, M.H. Eds. *Intelligent Interfaces Theory, Research and Design*. Elsevier Science Publishers B.V., 1989, 213-264.
- [GAIN86] Gaines, B.R. Foundations of dialog engineering: the development of human-computer interaction. Part II, *International Journal of Man-Machine Studies*, 24, 1986, 101-123.
- [GIT86] Gittens, D. Icon-Based Human-Computer Interaction *International Journal of Man-Machine Studies*, 24, 1986, 519-543.
- [GOLD84] Goldberg, A. *Smalltalk-80: The Interactive Programming Environment*. Xerox Corporation, 1984.
- [GOOD84] Good, M.D., Whiteside, J.A., Wixon, D.R. & Jones, S.J. Building a User-Derived Interface, *Communications of the ACM*, Vol 27, no 10, October 1984, 1032-1043
- [GRIC75] Grice, H.P. Logic and Conversation. In *Systems and Semantics: Speech Acts*, v.3, Cole P. & Morgan Eds., 1975.
- [HANC89] Hancock, P.A. & Chignell, M.H. Eds. *Intelligent Interfaces Theory, Research and Design*. Elsevier Science Publishers B.V., 1989.

- [HART89] Hartson, H.R. & Hix, D. Human-Computer Interface Development: Concepts and Systems for Its Management, *ACM Computing Surveys*, Vol.21, No.1, March 1989.
- [HIMM73] Himmelblau, D. (Ed.) Decomposition of large-scale systems. Amsterdam: Elsevier, 1973.
- [HUTC86] Hutchins, E.L., Hollan, D.J. & Norman, D.A. Direct Manipulation Interfaces. In Norman, D.A. & Draper, S.W. (Eds.), *User Centered System Design*, Hillsdale, N.J.: Lawrence Erlbaum Associates, 1986, 118-123.
- [JACOS89] Jacob, R.J.K. Direct Manipulation In The Intelligent Interface. In Hancock, P.A. & Chignell, M.H. (Eds.) *Intelligent Interfaces: theory, research and design*. Elsevier Science Publishers B.V., 1989, 165-212.
- [JAME86] Jameson, A. How to appear to be conforming to the "maxims" even if you prefer to violate them, *First draft for the Third International Workshop on Natural Language Generation*, August 1986.
- [KAIS88] Kaiser, G.E.; Feiler, P.H. & Popovich, S.S. Intelligent Assistance for Software Development and Maintenance, *IEEE Software*, May 1988, 40-49.
- [KAMM88] Kammersgaard, J. Four different perspectives on human-computer interaction, *International Journal of Man-Machine Studies*, 28, 1988, 343-362.
- [KANT89] Kantowitz, B.H. Interfacing Human and Machine Intelligence. In Hancock, P.A. & Chignell, M.H. (Eds.) *Intelligent Interfaces: theory, research and design*. Elsevier Science Publishers B.V., 1989, 49-68.
- [KENN88] Kennedy, A., Wilkes, A., Elder, L. & Murray, W.S. Dialogue with machines, *Cognition*, 30, 1988, pp.37-72.
- [LITMaa] Litman, D.J. & Allen, J.F. Discourse Processing and Commonsense Plans. Fonte e data ilegíveis.
- [LODD83] Lodding, K.N. Iconic Interfacing, *IEEE Computer Graphics and Applications*, 3(2), March/April 1983, 11-20
- [MANE85] Manes, S. Pushing Picture-Perfect Programs: Smash That Icon! *PC Magazine*, June 1985, 64.
- [MORA87] Moray, N. Intelligent aids, mental models, and the theory of machines, *International Journal of Man-Machine Studies*, 27, 1987, 619-629.
- [MUIR87] Muir, B.M. Trust between humans and machines, and the design of decision aids, *International Journal of Man-Machine Studies*, 27, 1987, 527-539
- [NADI87] Nadin, M. & Novak, M. MIND: A Design Machine – Conceptual Framework –, *Proc. 1st. EUROGRAPHICS Workshop on Intelligent CAD Systems – Theoretical and Methodological Aspects*, The Netherlands, April 21-24, 1987, 146-171.

- [NORM83] Norman, D.L. Design rules based on analyses of human error, *Communications of the ACM*, 26(4), 1983, 245-258.
- [NORM84] Norman, D.A. Stages and levels in human-machine interaction, *International Journal of Man-Machine Studies*, 21, 1984, 365-375.
- [NORM87] Norman, D.A. Some Observations on Mental Models. In Baecker, R.M. & Buxton, W.A.S. (writers and editors) *Readings in Human-Computer Interaction: A Multidisciplinary Approach*. Morgan Kaufmann Publishers, Inc., 1987, 241-244
- [OGBO83] Ogborn, J.M. & Johnson, L. Conversation Theory. *Kybernetes*, V.12, 1984, 7-16.
- [POTO88] Potosnak, K. Human Factors: tools, techniques, and concepts to optimize user interfaces, *IEEE Software*, May 1988, 97-99.
- [RICH83] Rich, E. Users are individuals: Individualizing user models, *International Journal of Man-Machine Studies*, 18, 1983, 199-214.
- [RICH84] Rich, E. Natural-Language Interfaces, *IEEE Computer*, Sept.1984, 39-47.
- [RICH88] Rich, E. *Inteligência Artificial*, Ed. McGraw-Hill, Ltda, São Paulo, 1988.
- [RISS84] Rissland, E. Ingredients of Intelligent User Interfaces, *International Journal of Man-Machine Studies*, 21, 1984, 377-388.
- [ROTH87] Roth, E.M., Bennett, K.B. & Woods, D.D. Human interaction with an "intelligent" machine, *International Journal of Man-Machine Studies*, 27, 1987, 479-525.
- [SHNE83] Shneiderman, B. Direct manipulation: A step beyond programming languages, *IEEE Computer*, 16, Aug. 1983, 57-69.
- [SHNE87] Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley Publishing Company, 1987
- [SIMP85] Simpson, C.A. et al. System Design for Speech Recognition and Generation. In Baecker, R.M. & Buxton, W.A.S. (writers and editors) *Readings in Human-Computer Interaction: A Multidisciplinary Approach*. Morgan Kaufmann Publishers, Inc., 1987, 400-413.
- [SZEK89] Szekely, P. Structuring Programs to Support Intelligent Interfaces, *ISI Reprint Series, ISI/RS-89-231*, University of Southern California, February 1989.
- [TAKA87] Takala, T. Intelligence beyond Expert Systems: A Physiological Model with Applications in Design, *Proc. 1st. EUROGRAPHICS Workshop on Intelligent CAD Systems - Theoretical and Methodological Aspects*, The Netherlands, April 21-24, 1987, 286-293.



- [TYLE89] Tyler, S.W. & Treu, S. An interface architecture to provide adaptive task-specific context for the user, *International Journal of Man-Machine Studies*, 30, 1989, 303-327.
- [VASS84] Vassilou, Y.(Ed.) Human Factors and Interactive Computer Systems. Ablex Publishing Corporation, 1984.
- [WATE86] Waterman, D.A. A Guide to Expert Systems. Addison-Wesley Publishing Company, Inc. 1986.
- [WENG87] Wenger, E. Artificial Intelligence and Tutoring Systems: computational and cognitive approaches to the communication of knowledge. Morgan Kaufmann Publishers, Inc. 1987
- [WINS84] Winston, P.H. Artificial Intelligence. Addison Wesley Publishing Company, Inc. 1984.