

PATRICIA RUCKER DE BASSI

**UM AMBIENTE DE INTERFACE VISUAL PARA A  
GERAÇÃO DE SUB-ESQUEMAS PARA UMA  
FERRAMENTA DE CONSULTA BASEADA  
NO MODELO ERC<sup>+</sup>**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática, Curso de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Prof.<sup>ª</sup> Dr.<sup>ª</sup> Laura Sánchez García

CURITIBA

2002



Ministério da Educação  
Universidade Federal do Paraná  
Mestrado em Informática

## PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática da aluna *Patícia Rucker de Bassi*, avaliamos o trabalho intitulado "*Um Ambiente de Interface Visual para a Geração de Sub-Esquemas para uma Ferramenta de Consulta Baseada no Modelo ERC+*", cuja defesa foi realizada no dia 27 de fevereiro de 2002, às quatorze horas e trinta minutos, no anfiteatro A do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação da candidata.

Curitiba, 27 de Fevereiro de 2002.

Prof.<sup>a</sup> Dra. Laura Sanchez Garcia  
PGINF/UFPR - Orientadora

Prof. Dr. Osvaldo Luiz de Oliveira  
USF

Prof. Dr. Marcos Sfair Sunye  
DINF/UFPR

Esta dissertação é dedicada aos meus pais Mario e Guerrit que me deram toda a base educacional para que eu pudesse realizar esta pesquisa. Em especial ao meu marido Danilo pela infundável paciência e apoio, é muito bom ter você ao meu lado.

Também dedico este trabalho em especial à minha "mestre" e incentivadora Laura Sanchez Garcia, pois sem seu crédito, atenção e competência, este meu objetivo não seria realizado.

## AGRADECIMENTOS

Inicialmente gostaria de agradecer aos meus amigos, onde tudo começou: Cristiani Neves Batata, Denis Alcides Rezende, Andrea Rodacki, Fábio Dias Araújo e demais colegas de aula.

Gostaria de agradecer também aos amigos que fiz durante o desenrolar deste curso e que dividiram comigo a descoberta deste conhecimento: Clodis Boscaroli, Ezequiel Gueiber e Geraldo Ranthum.

Aos professores Deborah Ribeiro Carvalho, Luiz Carlos Klug, Luiz Roberto Baracho Rocha, Marcos Antonio Canalli, Ricardo Oliveira Pereira, Tânia Martins Preto, pela ajuda constante, seja na escrita da dissertação ou nas atividades do cotidiano.

A Universidade Federal do Paraná e a todo Departamento de Informática, professores e funcionários. Em especial, ao Prof. Marcos Sunyé pelas contribuições, esclarecimentos e pelo oferecimento do espaço de pesquisa, e ao Prof. Alexandre Direne pela motivação e contribuições que muito enriqueceram este trabalho.

Aos meus amigos e familiares que sempre me deram apoio nas horas de estudo.

E finalmente, gostaria de agradecer a Deus pela suas energias que inspiram minha serenidade e consciência.

# SUMÁRIO

LISTA DE ILUSTRAÇÕES .....	VII
LISTA DE ABREVIATURAS E SIGLAS .....	IX
RESUMO .....	X
ABSTRACT .....	XI
1. INTRODUÇÃO .....	1
2. DESIGN E AVALIAÇÃO DE INTERFACES HUMANO-COMPUTADOR.....	6
2.1 BASES TEÓRICAS.....	6
2.1.2 Engenharia Cognitiva .....	10
2.1.2 Engenharia Semiótica .....	11
2.2 INTERFACE DE MANIPULAÇÃO DIRETA .....	13
2.3 LINGUAGENS VISUAIS.....	17
2.4 DESIGN DE INTERFACES HUMANO-COMPUTADOR .....	20
2.4.1 Auxílio ao Usuário.....	24
2.5 AVALIAÇÃO DE INTERFACES HUMANO-COMPUTADOR .....	26
2.5.1 Linguagem de Especificação da Mensagem do Projetista .....	29
2.5.2 Escolha do Método Semiótico de Análise .....	34
3. METODOLOGIA PARA CONSTRUÇÃO DA INTERFACE.....	36
3.1 DESCRIÇÃO DA METODOLOGIA .....	36
3.1.1 Análise de Usuários e Tarefas .....	38
3.1.2 Mapeamento Tarefa-Ação.....	38
3.1.3 Modelagem de Tarefas.....	39
3.1.3.1 <i>Task-Action Grammar</i> – Gramática Tarefa-Ação .....	40
3.1.3.2 <i>User Action Notation</i> – Notação da Ação do Usuário .....	41
3.1.3.3 <i>Goals, Operators, Methods and Selection Rules</i> – Metas, Operadores, Métodos e Regras de Seleção.....	41
3.1.3.4 Escolha do método de modelagem de tarefas .....	41
3.1.4 Considerações sobre os Modelos de Tarefas .....	45
4. FERRAMENTAS DE CONSULTA A BASE DE DADOS.....	48
4.1 O MODELO DE DADOS.....	48
4.1.1 O Modelo ERC+ .....	50
4.1.2 A Álgebra do Modelo ERC+.....	55

<b>4.2 FORMULAÇÃO DE UMA CONSULTA EM UMA BASE DE DADOS.....</b>	<b>59</b>
<b>4.3 A INTERFACE NO PROCESSO DE CONSULTA EM UMA BASE DE DADOS</b>	<b>61</b>
<b>4.4 PANORAMA GERAL DE FERRAMENTAS DE CONSULTAS A BASE DE DADOS .....</b>	<b>63</b>
<b>4.5 ANÁLISE DAS FERRAMENTAS PASTA3, QBD* E SUPER.....</b>	<b>65</b>
4.5.1 Pasta-3.....	65
4.5.2 QBD* .....	68
4.5.3 Super .....	70
<b>5. A FERRAMENTA VIQUEN .....</b>	<b>73</b>
<b>5.1 DESCRIÇÃO DA VIQUEN .....</b>	<b>73</b>
<b>5.2 AVALIAÇÃO DA INTERFACE DA FERRAMENTA VIQUEN .....</b>	<b>78</b>
5.2.1 Avaliação com base nas Medidas Semióticas.....	81
5.2.2 Conclusão da Avaliação.....	86
<b>6. A INTERFACE PARA A SELEÇÃO DO SUB-ESQUEMA.....</b>	<b>88</b>
<b>6.1 CARACTERIZAÇÃO DA INTERFACE .....</b>	<b>88</b>
<b>6.2 A METODOLOGIA PARA A ESPECIFICAÇÃO DA INTERFACE.....</b>	<b>93</b>
6.2.1 Análise do Perfil do Usuário.....	94
6.2.2 Análise de Tarefas .....	94
<b>6.3 A APLICAÇÃO .....</b>	<b>95</b>
6.3.1 As Etapas da Aplicação .....	95
6.3.2 As Mensagens da Aplicação .....	103
6.3.3 O Auxílio ao Usuário.....	109
6.3.4 A LEMD da interface proposta .....	111
6.3.5 A STAG da interface proposta.....	111
6.3.6 O Domínio da Aplicação .....	112
6.3.7 Exemplo de Consulta.....	114
6.3.8 O Auxílio <i>Online</i> .....	123
6.3.9 Comparação com a Ferramenta VIQUEN.....	124
<b>7. CONCLUSÃO .....</b>	<b>127</b>
<b>8. TRABALHOS FUTUROS .....</b>	<b>130</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>131</b>
<b>APÊNDICE 1 – A LEMD DA INTERFACE .....</b>	<b>137</b>
<b>APÊNDICE 2 – A STAG DA INTERFACE.....</b>	<b>151</b>



## LISTA DE ILUSTRAÇÕES

Figura 1: Interseção Humano-computador.....	7
Figura 2: Ciclo cognitivo na interação usuário-sistema.....	11
Figura 3: Modelos da aplicação.....	12
Figura 4: Espaço da Interface definindo as categorias de linguagens.....	19
Tabela 1: Exemplos de regras de mapeamento semântico.....	32
Figura 5: O processo de design de interfaces na Engenharia Semiótica.....	37
Figura 6: Formalismos de apoio ao design.....	47
Figura 7: Objetos representados de acordo com o modelo ERC+.....	53
Figura 8: Arquitetura de um ambiente de VQS baseado em Angelaccio et al., 1990 (b).....	63
Figura 9: Esquema de uma propriedade agrícola sob o modelo ERC+.....	75
Figura 10: Seleção de atributos sob o modelo ERC+.....	75
Figura 11: Exemplo de uma consulta representada por um grafo acíclico com raiz escolhida.....	76
Figura 12: Subesquema proprietário adicionado a consulta.....	76
Figura 13: Consulta após a unificação dos subesquemas.....	76
Figura 14: Seleção da complementação do predicado.....	77
Figura 15: Representação do predicado em forma de árvore hierárquica e SQL.....	77
Figura 16: Resposta da consulta mantendo a estrutura de objetos e representação hierárquica em forma de árvore.....	78
Tabela 2: Dicionário de signos do VIQUEN.....	79
Tabela 3: Tarefas (Etapas) do processo de consulta ao VIQUEN.....	80
Tabela 4: Textos das telas.....	103
Tabela 5: Texto de ajuda.....	105
Tabela 6: Texto de mensagens de sub-etapas críticas.....	107
Figura 17: Diagrama ERC+ do domínio da aplicação.....	113
Figura 18: Tela inicial da interface.....	115
Figura 19: Apresentação do Esquema Original no modelo ERC+.....	115
Figura 20: Seleção da raiz da consulta.....	116
Figura 21: Esquema original utilizando a metáfora do sujeito da sentença para a raiz da consulta..	116
Figura 22: Seleção/remoção dos objetos que compõem o predicado da sentença, que está sendo utilizada como metáfora da consulta.....	117
Figura 23: Apresentação do sub-esquema construído , para eliminação de uma estrutura cíclica – Pesquisador/Realiza/Trabalho/Publica/Trabalho.....	117
Figura 24: Apresentação da seleção de uma ligação para a abertura da estrutura cíclica.....	118
Figura 25: Apresentação do sub-esquema sem ciclos para adição de novos elementos.....	118
Figura 26: Apresentação da tela para seleção de novos objetos, onde está sendo mostrado o esquema original com os objetos já selecionados para a seleção de novos objetos.....	119



Figura 27: Apresentação da tela mostrando a seleção de um novo objeto, além dos já selecionados anteriormente.....	119
Figura 28: Apresentação de tela mostrando a opção de adição de um novo sub-esquema onde serão apresentados os arquivos já gravados de sub-esquemas utilizando a mesma base de dados do esquema original em uso. ....	120
Figura 29: Apresentação de tela mostrando o sub-esquema já selecionado, junto ao novo objeto selecionado para unificação.....	120
Figura 30: Apresentação de tela mostrando a seleção de um objeto duplicado nos dois sub-esquemas, onde será feita a união.....	121
Figura 31: Apresentação da tela mostrando o sub-esquema unificado para seleção dos atributos que comporão a consulta.....	121
Figura 32: Apresentação de tela mostrando a seleção de um objeto para inclusão de seus atributos na consulta. ....	122
Figura 33: Apresentação de tela mostrando a seleção dos atributos do objeto selecionado. ....	122
Figura 34: Apresentação de tela finalizando a etapa de construção do sub-esquema, indicando a passagem para outra etapa do processo de construção da consulta. ....	122

## LISTA DE ABREVIATURAS E SIGLAS

- D-TAG - extensão da *Task-Action Grammar*
- DBMS - *Data Base Management System*
- E-R - Entidade Relacionamento
- ERC+ - Modelo Entidade Relacionamento Complexo Estendido
- ERC - Modelo Entidade Relacionamento Complexo
- ES - Engenharia Semiótica
- GOMS - *Goals, Operators, Methods and Selection Rules*
- GrDM – *Graphic Direct Manipulation*
- GUI - *Graphical Users Interface*
- IHC - Interação Humano-Computador
- LDIU - Linguagem de Design de Interface de Usuário
- LEMD - Linguagem de Especificação da Mensagem do Projetista
- LV - Linguagem Visual
- MD - Manipulação Direta
- OO - Orientação a Objetos
- QBD\* - *Query By Diagrams*
- QBE - *Query By Example*
- SGBD - Sistema Gerenciador de Banco de Dados
- SQL - *Structured Query Language*
- STAG - *Semiotic Task-Action Grammar*
- TAG - *Task-Action Grammar*
- UAN - *User Action Notation*
- UNICAMP – Universidade Estadual de Campinas
- UFPR – Universidade Federal do Paraná
- VQL - *Visual Query Language*
- VQS - *Visual Query Systems*
- VIQUEN - *Visual Query Environment*
- WIMP - *Windows, Icons, Menus and Pointer devices*
- WYSIWYG - “*What you see is what you get*”

## RESUMO

Este trabalho apresenta um protótipo de uma interface para construção de sub-esquema em uma ferramenta de consulta visual a banco de dados visando o usuário com pouco conhecimento em conceitos de banco de dados mas com conhecimento no ambiente Windows e no domínio da base de dados em questão. Faz uma revisão sobre o estudo de Interface Humano-Computador, particularmente sobre a Engenharia Semiótica e Engenharia Cognitiva e revisa metodologias de design e avaliação de interfaces. Recuperando os passos metodológicos, relata a análise de ferramentas visuais de consulta a base de dados existentes no meio acadêmico e, em particular, do ambiente VIQUEN, buscando ressaltar os pontos positivos e negativos das ferramentas à luz dos conceitos de Interação Humano-Computador. Apresenta o desenvolvimento de uma proposta de interface para o ambiente VIQUEN, seguindo os passos da metodologia de design de interface escolhida, fundamentada nos conceitos de Engenharia Semiótica e mais especificamente numa linguagem de especificação de design de interface, a LEMD, e de uma técnica de geração de linguagem visual, a STAG, considerando também os recursos do auxílio ao usuário em função da complexidade técnica do domínio escolhido. Por fim, o trabalho apresenta metas futuras de pesquisa que consistem basicamente na adaptação da interface proposta ao ambiente VIQUEN e na fusão das três etapas da construção da consulta – sub-esquema, predicado e resposta – numa ferramenta de consulta visual a banco de dados para ser utilizada por usuários com pouca experiência em conceitos técnicos da área.

**Palavras-chave:** Interação Humano-Computador, Linguagens de Consultas Visuais a Banco de Dados.

## ABSTRACT

This work presents interface prototype for subschema construction in a visual query data base tool. To support novice users in data base concepts, with experience in Windows environment and the data base domain. It makes a revision about Human-Computer Interface mainly about Semiotic Engineering and Cognitive Engineering and reviews methodologies of design and interface evaluation. Retrieving the methodology steps reports to visual query data base tool analyses available in academic environment, and particularly the VIQUEN tool. In order to emphasizes the positive and negative items under the Human-Computer Interface concepts. Presenting the development of an interface proposal for the VIQUEN environment, following the chosen methodology of interface design, which is based on Semiotic Engineering concepts and more specifically on an Interface Design Specification Language, the LEMD, and a visual language generation technique, the STAG, and also considering the users assistance, because of the technical difficulty of the chosen domain. Finally, this work presents future goals and research, which consists basically on the adaptation of the proposed VIQUEN interface and the connection of the query construction three steps - subschema, predicate and results – in a visual query data base tool in order to be used by novice users in technical data base concepts.

Key-words: Human-Computer Interaction, Data Base Visual Query Language.

## 1. INTRODUÇÃO

Enquanto os bancos de dados são projetados, criados e modificados por profissionais da área de Sistemas de Informação, usuários de outras áreas do conhecimento acessam estas bases de dados para extrair informações relevantes para suas tarefas. Para que tal objetivo seja alcançado foram definidas linguagens específicas, chamadas linguagens de consulta. Uma linguagem de consulta consiste em um conjunto de operadores formais que permite a realização de perguntas à base de dados, que devem seguir uma sintaxe rigorosa. O sistema produz resultados que são extraídos dos dados armazenados, respondendo de forma coerente à consulta realizada.

As linguagens tradicionais de consulta não auxiliam no entendimento do significado dos dados, nem procuram entender as necessidades do usuário. De uma forma geral, as interfaces de linguagens de consultas (tais como SQL - *Structured Query Language*) ainda não estão embasadas nos requisitos de uma boa interface do ponto de vista da Interação Humano-Computador (IHC). Hoje em dia, ao invés de o usuário utilizar estas linguagens de consulta que necessitam de um conhecimento prévio da área de Sistemas de Informação, tipicamente conceitos de sintaxe dos comandos a serem utilizados e informações tanto do domínio da aplicação em questão quanto de seus mecanismos de interação, ele pode se aproveitar de interfaces gráficas que procuram aproximar a linguagem do sistema à realidade do usuário.

Com o crescimento dos usuários de bancos de dados tem se acentuado o desenvolvimento de interfaces visando facilitar o processo de interação entre o homem e o computador (Angelaccio et al., 1990 (b)). Dentre elas pode ser citada a

que utiliza linguagem de manipulação direta, caracterizada pela visualização dos objetos de interesse e a substituição da linguagem conversacional pela Manipulação Direta (MD) dos objetos da base de dados, representados visualmente. Este modelo de interação é analisado nos estudos de Hutchins, Hollan e Norman (Hutchins et al., 1986).

Tem-se conhecimento de um grupo de estudiosos na área de Banco de Dados que estão voltados para pesquisar e desenvolver ferramentas que permitam a usuários leigos entender e extrair de forma “simples” da base de dados as informações para o problema em questão. As linguagens de quarta geração, mesmo que não procedurais, não são consideradas suficientemente apropriadas para usuários leigos, que precisam ter o conhecimento da estrutura lógica do banco de dados e da sintaxe e semântica das linguagens dos SGBDs (Sistemas Gerenciadores de Banco de Dados). Entretanto, as propostas recentes de linguagens visuais que permitem aos usuários extrair informações por meio de interação e manipulação direta, ainda não são capazes de combinar, de forma eficiente, a facilidade de uso e o alto poder de expressão, que são necessários para criar as consultas desejadas pelos usuários. (Catarci & Santucci, 1995)

Geralmente, quando se faz uso de manipulação direta dos objetos, estes objetos são símbolos gráficos que devem ter algum significado no mundo real do usuário. Quando se faz uso de símbolos, ou signos na Engenharia Semiótica (ES) (De Souza et al., 1999 (a)), para representar situações que sejam comuns aos usuários, está sendo utilizado o recurso de metáfora. As metáforas são exploradas para representar a parte intencional da base de dados, isto é, o projeto de dados. Nestes casos, a metáfora é o mediador entre o modelo de dados e o usuário. Entretanto, o usuário tem um certo modelo mental de sua realidade de interesse

(Norman, 1986), a qual não necessariamente é refletida pela metáfora utilizada para representar o modelo de dados. Os projetistas de interfaces de sistemas de base de dados têm explorado metáforas que são facilmente entendidas por usuários experientes, tal como metáforas de tabelas ou nodos e arcos, mas não necessariamente pelos usuários leigos (Catarci et al., 1995).

Desta forma, a necessidade de uma boa interação entre usuários e sistemas de banco de dados tem sido reconhecida e discutida na comunidade acadêmica, levando a maioria das linguagens de consulta a evoluir em dois eixos: capacidade funcional e usabilidade (Catarci et al., 1997). A primeira está relacionada com o poder da linguagem, isto é com o que o usuário é capaz de fazer utilizando a linguagem, e as alternativas de que o usuário dispõe para apresentar as respostas. A segunda diz respeito ao esforço cognitivo, isto é, a quanto esforço o usuário necessita para interagir com o sistema, que vem ao encontro dos resultados dos estudos de Norman ilustrados pelos conceitos de golfos de execução e avaliação (Norman, 1986).

Sob este ponto de vista, o estudo da interação humano-computador envolve conhecimento sobre o humano por um lado, sobre a tecnologia por outro e sobre as maneiras como um influencia e é influenciado pelo outro. Isto leva ao fato de que o processo de design em IHC tem incorporado questões relativas a modelos cognitivos do processamento humano e tem sido naturalmente centrado no usuário. Por este motivo o processo de design deve ser analisado tanto pela perspectiva da Engenharia Cognitiva como da Engenharia Semiótica, que aponta propostas de linguagens de design para interfaces e está baseada em teorias fundamentadas no paradigma da comunicação, além dos princípios de Design Centrado no Usuário.

No âmbito desta evolução, surgiram modelos de dados que viabilizaram o enriquecimento da representação de objetos do mundo real, facilitando a sua manipulação pelo usuário. Dentre estes modelos está o ERC+ (modelo relacional estendido com tratamento de objetos complexos). Também com este propósito, um grande número de ambientes de consulta para bases de dados tem sido proposto com esta finalidade, dentre as quais está o VIQUEN (*Visual Query Environment*) (Gueiber, 2001), que é um ambiente interativo para consulta visual e extração de esquemas de uma base de dados, uma ferramenta construída para o modelo ERC+. Embora as vantagens destes ambientes visuais sejam amplamente ressaltadas na literatura, são necessários, ainda, estudos mais aprofundados capazes de atestar a real facilidade potencial oferecida por este tipo de ambiente. Este ambiente foi escolhido como ferramenta de partida para o desenvolvimento desta pesquisa.

Este trabalho se propõe a realizar um diagnóstico das linguagens de consultas visuais a banco de dados disponíveis na literatura, apontando para os aspectos que ainda exigem um esforço adicional por parte do usuário. Mais precisamente serão objetos do presente estudo linguagens visuais que visam o tratamento prévio da consulta, permitindo a geração do sub-esquema. Como principal contribuição para este trabalho, emergirá a proposta de um ambiente de interface para a geração do sub-esquema, baseando no protótipo VIQUEN.

A presente dissertação está organizada da seguinte forma: a seção 2 trata das teorias subjacentes à área IHC, abordando também o processo de design de interfaces e uma metodologia de avaliação semiótica de interfaces; a seção 3 descreve a metodologia de construção de interface proposta por De Souza (De Souza et al., 1999 (a)). Já a seção 4 descreve conceitos relativos à área de Banco de Dados dando ênfase ao processo de consulta e à análise de ferramentas



acadêmicas de consulta visual a banco de dados, levantando seus pontos positivos e negativos à luz dos conceitos de IHC. A ferramenta VIQUEN é descrita e analisada, com base no método de avaliação semiótica de interfaces, na seção 5. A interface proposta para o VIQUEN, que leva em consideração a metodologia de design de interface escolhida e os itens relevantes da avaliação realizada na ferramenta em questão, é descrita na seção 6. Por fim, a conclusão alcançada e as propostas de trabalhos futuros decorrentes do estudo realizado são discutidas nas seções 7 e 8, respectivamente.

## **2. DESIGN E AVALIAÇÃO DE INTERFACES HUMANO-COMPUTADOR**

Neste capítulo é apresentado um detalhamento dos pressupostos teóricos sobre os quais este trabalho foi construído. Atendendo a esta necessidade, dois grandes eixos teóricos foram explorados, a abordagem semiótica, no que se refere à produção sógnica, a abordagem cognitiva, no que se refere às questões de percepção e compreensão. Será apresentado também o pressuposto de que a regularidade da linguagem é um elemento importante no que se refere à sua capacidade expressiva, logo a especificação de linguagens visuais de forma sistemática necessita de uma abordagem especial de forma a oferecer um tratamento adequado às especificidades dos recursos visuais. Também o desafio de usabilidade dos artefatos de software aplicativo trouxe a necessidade da associação de fundamentos teóricos com a prática de design, o que conduziu a um processo de design baseado em diretrizes e princípios que está sendo relatado neste capítulo. A partir destes princípios que determinaram a criação de uma linguagem de especificação de design de interface foi possível a utilização desta mesma linguagem em um processo de avaliação de interface, como segue relatado.

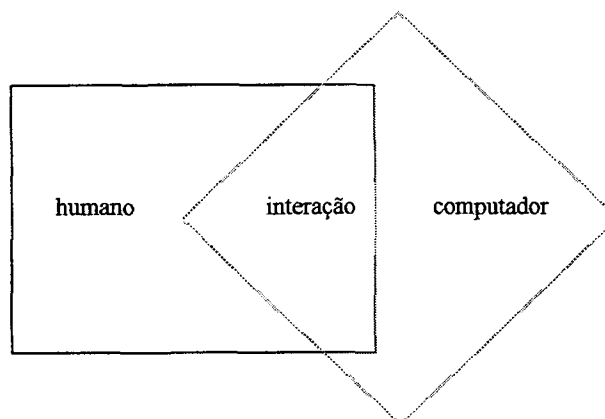
### **2.1 BASES TEÓRICAS**

Os primeiros trinta anos da história dos computadores foram marcados pelo design centrado na tecnologia: as pessoas devendo se ajustar a uma perspectiva centrada na máquina, uma vez que era uma ferramenta de domínio exclusivo de especialistas. Nos anos oitenta, o design de software passou por uma mudança de paradigma: do foco centrado na tecnologia para a perspectiva do usuário

(Baranauskas et al., 1998). Nesta nova abordagem Norman em (Norman, 1986) cita que *as necessidades do usuário devem dominar o design da interface, e as necessidades da interface devem dominar o design do restante do sistema.*

O design de muitos objetos que encontramos diariamente – uma xícara, uma placa de sinalização de trânsito – não são aleatórios. Para que o ser humano possa usar efetivamente seu ambiente, suas ferramentas, suas máquinas, o design delas precisa ser apropriado às situações de uso. A xícara, diferentemente de um copo, tem alça, pois é utilizada para servir bebidas quentes, de outra forma queimaríamos a mão. Na mesma medida, para que computadores sejam usados efetivamente, o design de seus softwares precisam ser apropriados. Design de software é um campo relativamente novo que estuda a interseção entre humanos e computador (figura 1) e as várias interfaces (física, sensoriais, psicológicas) que os conectam. Segundo esta visão o papel do computador é o de um *medium* – uma substância física na qual signos são manifestados e usados para comunicação (Oliveira & Baranauskas, 1998 (a)).

Figura 1: Interseção Humano-computador



Fonte: Nadin, 2000

Desta forma, segundo Oliveira e Baranauskas em (Oliveira & Baranauskas, 1998 (a)), a interface de software pode ser entendida como um sistema de signos manifestados em processos computacionais, que as pessoas conscientes ou inconscientemente criam quando usam ou interpretam sistemas computacionais.

De Souza et al. em (De Souza et al., 1999 (a)) afirmam que, na indústria de software, o design de interface tem sido conduzido por meio de processos iterativos de construção e avaliação de protótipos baseados em princípios e diretrizes empíricas. Porém a execução desta atividade tem-se deparado com conflitos entre os conceitos apresentados em determinadas situações. Para resolvê-los, tornou-se necessário basear a prática de design de interfaces em uma fundamentação teórica. Esta fundamentação teórica objetiva orientar o projetista ao longo da elaboração da sua solução particular para o conjunto de problemas que a aplicação pretende resolver. Com teorias a respeito dos fenômenos envolvidos é possível prever antecipadamente se o sistema a ser desenvolvido irá satisfazer aos requisitos de usabilidade e comunicabilidade, oferecendo aos usuários artefatos de uso e comunicação fáceis.

A abordagem tradicional que tem caracterizado a pesquisa da IHC e, em particular, o design de software, tem sido a cognitiva. Cognição refere-se ao processo pelo qual as pessoas se tornam conscientes das coisas, incluindo aí o entendimento, capacidade de memória, raciocínio, atenção, aquisição de habilidades e criação de novas idéias, em suma adquirir conhecimento. A principal meta de pesquisa nesta área do conhecimento tem sido entender como os seres humanos interagem com os computadores em termos da forma como o conhecimento é transmitido entre os dois (Oliveira & Baranauskas, 1998 (a)). Assim, para praticar

design é preciso pensar e criar estruturas que dêem suporte à inter-relação Humano-Computador, conforme figura 1.

Quando se desenvolve um software o dialeto que a interface do computador irá utilizar deverá ser composto do dialeto de trabalho do usuário mais um corpo de signos residentes na interface. Desta forma, o software deve existir dentro de um contexto de valores e necessidades das pessoas que irão utilizá-lo. A abordagem puramente cognitiva não provê uma base apropriada para a consideração de fenômenos de natureza lingüística e de interação de grupos de pessoas. Em conseqüência, estudos sobre a interação Humano-Computador devem levar em conta aspectos subjacentes ao contexto cultural e social em que a interação se dá, esta abordagem é tratada pela Semiótica aplicada ao design de software (Oliveira & Baranauskas, 1998 (a)).

Segundo Prado e Baranauskas em (Prado & Baranauskas, 2000), a avaliação de softwares tem sido historicamente associada a julgamentos de valor, realizados em etapas finais do desenvolvimento, que conduzem a uma aprovação ou reprovação com relação a determinadas características. O entendimento da natureza cíclica do processo de design, entretanto, tem conduzido a um novo conceito para o que significa avaliar artefatos de software e, em particular, avaliar software. Este fato tem trazido o foco das técnicas e métodos de avaliação para dentro do próprio processo de design, gerando uma nova fase no processo. Várias metodologias de análise de interfaces comprovadas academicamente utilizam os conceitos oriundos de Cognição e Semiótica, e buscam avaliar o quanto os sistemas estão mais próximos do usuário e a maximização da correspondência entre o sinal e seu referente, respectivamente. Isto permite um exame detalhado das interfaces

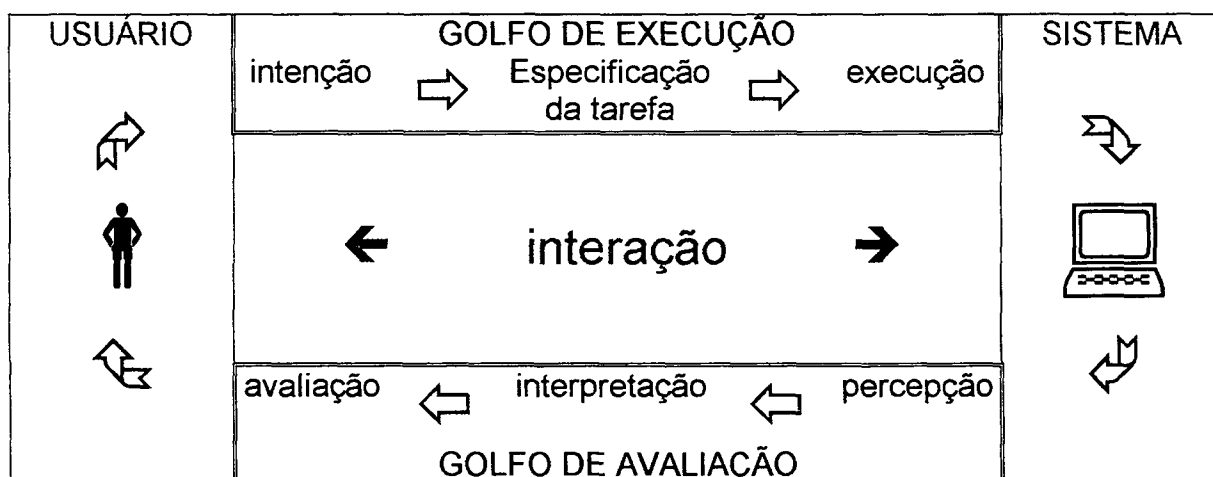
disponíveis no mercado revelando que esses princípios não têm sido respeitados (Baranauskas et al., 1998).

### 2.1.2 Engenharia Cognitiva

A Engenharia Cognitiva, conforme proposto por Norman (Norman, 1986), tem raízes comuns às áreas de Psicologia Cognitiva, Ciência Cognitiva e Inteligência Artificial que estudam a cognição, isto é, o processo pelo qual se pode adquirir conhecimento, e aplicam suas teorias na compreensão das capacidades e limitações da mente dos usuários. A idéia básica da estratégia de abordagens cognitivas é que modelos cognitivos que descrevem os processos e estruturas mentais (tais como recordação, interpretação, planejamento e aprendizado) podem indicar para pesquisadores e projetistas quais as propriedades que os modelos de interação devem ter de maneira que a interação possa ser desempenhada mais facilmente pelos usuários (De Souza et al., 1999 (a)). Esta abordagem considera que o projetista inicialmente cria o seu modelo mental do sistema, chamado modelo de design, com base nos modelos de usuário e tarefa. O modelo implementado deste modelo de design é a imagem do sistema. O usuário então interage com esta imagem do sistema e cria seu modelo mental da aplicação, chamado de modelo do usuário. Este modelo mental é que permite ao usuário formular suas intenções e objetivos em termos de comandos e funções do sistema (De Souza et al., 1999 (a)). Para que isto seja possível, Norman argumenta (Norman, 1986) que o projetista precisa entender o processo por meio do qual o usuário interage com a interface do sistema o que se explica pela teoria da ação. A teoria da ação define que a interação usuário-sistema é desempenhada num ciclo-de-ação com etapas e “golfos” a serem atravessados: o golfo de execução, onde o usuário formula suas intenções sobre a

interface, traduzindo-as em ações do sistema; e o golfo de avaliação, onde o usuário avalia o resultado apresentado pelo sistema em relação ao seu objetivo (figura 2) (Norman, 1986).

Figura 2: Ciclo cognitivo na interação usuário-sistema



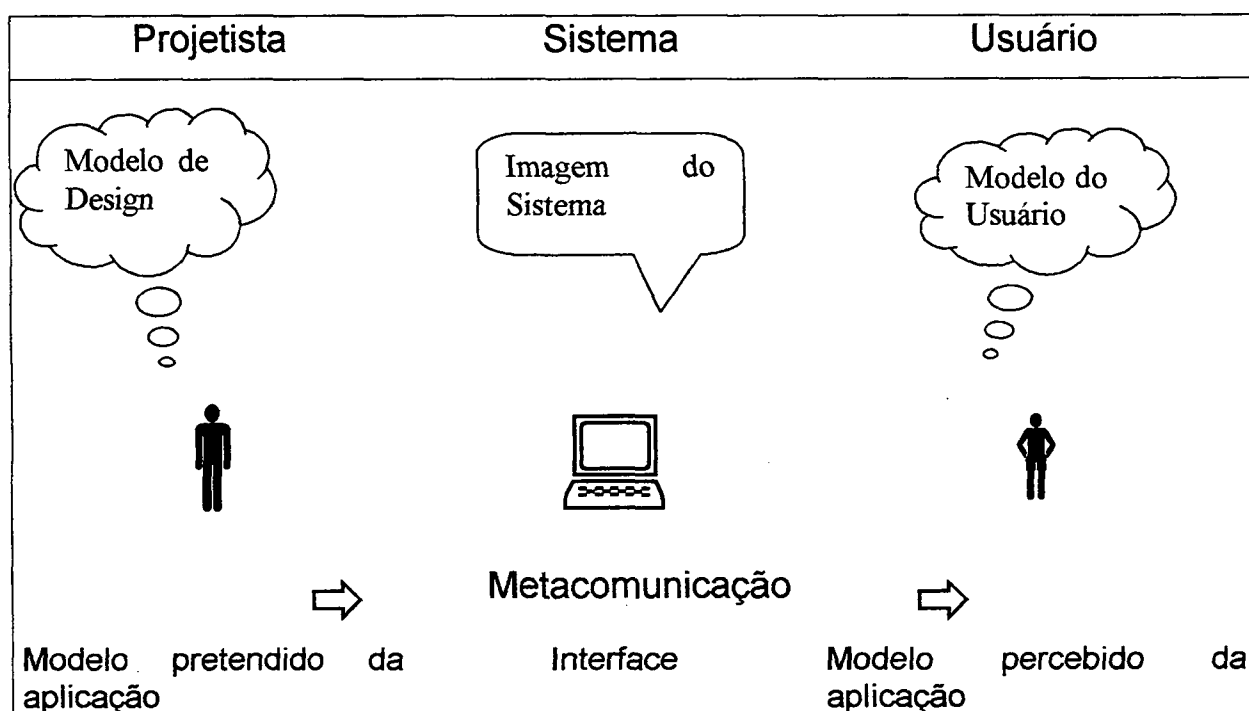
Fonte: Norman, 1986

### 2.1.2 Engenharia Semiótica

Já a abordagem da Engenharia Semiótica (De Souza, 1993) tem como base teórica a Semiótica, disciplina que estuda os signos, os sistemas semióticos e de comunicação, bem como os processos envolvidos na produção e interpretação de signos. Signo é algo que representa alguma coisa para alguém, qualquer marca, movimento físico, símbolo, sinal, entre outros, usado para indicar e veicular pensamentos, informação e comandos; constitui um signo (Oliveira & Baranauskas, 1999). Nesta abordagem toda aplicação computacional é concebida como um ato de comunicação que inclui o projetista no papel de emissor de uma mensagem para os usuários dos sistemas por ele criados (De Souza, 1993). A característica desta abordagem semiótica é que os artefatos de software aplicativo são artefatos de meta-comunicação (figura 3). Eles são considerados mensagens unilaterais

enviadas do projetista para o usuário sobre o domínio que o usuário detém sobre o sistema para obter o efeito desejado. Esta visão é complementar ao modelo clássico de design centrado no usuário de IHC, baseada na metáfora de Norman dos golfos que o usuário deve atravessar para interagir com o sistema: golfo da execução e golfo da avaliação. Esta nova abordagem caracteriza IHC como tendo dois níveis de comunicação: a comunicação projetista-usuário e a interação usuário-sistema. A interação humano-computador somente será atingida quando a comunicação for atingida nos dois níveis. Isto ocorre quando o usuário entende a mensagem principal do projetista sobre o tipo de mensagem que o sistema é capaz de interpretar, assim a interface parece intuitiva, útil e consistente ao usuário, e não propriedade de uma ou outra escolha de símbolos de interface (*widgets*, figuras, palavras, leiaute ou estruturas de diálogo) (De Souza et al., 2000).

Figura 3: Modelos da aplicação



Fonte: De Souza et al., 1999 (a)



Como foi recém visto, tanto a Engenharia Semiótica quanto a Engenharia Cognitiva enxergam o processo de design iniciando-se com o projetista que cria o modelo mental da aplicação, e com base neste, implementa a própria aplicação. O usuário interage com esta aplicação e por meio dela cria o seu próprio modelo mental da aplicação. A criação da aplicação pelo projetista e a interação do usuário são assíncronas, ou seja, se dão em diferentes momentos no tempo (De Souza et al., 1999 (a)).

O sistema de códigos expressivos do usuário no ambiente do computador é bastante simples (recursos do mouse), sua capacidade expressiva depende em grande parte da utilização de metáforas no ambiente de interação. A noção de metáfora em interface foi criada para descrever a correspondência entre o que o usuário vê na interface e o que ele deve pensar sobre o significado do que ele vê. Ao invés de pensar no próprio sistema representado na interface, a metáfora coloca junto o sistema e um domínio familiar ao usuário. O efeito disto é que o usuário irá desenvolver um modelo mental do sistema que estará mais próximo do mundo da metáfora do que do mundo do sistema representado (Oliveira & Baranauskas, 1999).

## 2.2 INTERFACE DE MANIPULAÇÃO DIRETA

Na busca pelo uso de metáforas do mundo real procurando aproximar o sistema computacional do modelo de tarefas do usuário há pelo menos vinte anos vem-se procurando utilizar recursos que permitam representar estas metáforas, porém seu desenvolvimento ocorreu faz pouco tempo com a melhoria dos recursos de hardware que permitem suportar a programação gráfica, dando mais poder para a representação do mundo real na tela do computador. Este recurso foi denominado por Schneiderman de Manipulação Direta e tem como objetivo referenciar interfaces

contendo as seguintes propriedades: representação contínua do objeto de interesse, ações físicas ou pressão de botões com rótulos ao invés de sintaxes complexas e mudança de estado rápido do objeto de interesse após a ação do usuário cujo impacto seja imediatamente percebido pelo usuário (Hutchins et al. 1986).

A principal metáfora implícita na interação por MD é a utilização do mouse como uma extensão da mão do usuário. Quando o usuário aponta para um objeto na tela, posicionando o cursor por meio do mouse, a analogia é de que ele estaria apontando com o próprio dedo, gerando o que é chamado de “manipulação direta” (Martins & De Souza, 1998).

Na MD os processos analógicos são utilizados em larga escala, de tal forma que a análise do sistema de códigos expressivos do usuário não pode ser feita desconsiderando a presença de metáforas. De um modo geral, a utilização de metáforas permite que uma quantidade menor de informação precise ser explícita. Quando o usuário captura a metáfora utilizada, de forma consciente ou intuitiva, ele transporta sua experiência em ambiente normal para o ambiente computacional (Martins & De Souza, 1998).

A promessa da MD é de que ao invés de se ter um meio computacional abstrato, toda a programação é feita graficamente, de uma forma que se aproxima da forma como as pessoas imaginam o problema. As operações desejadas são executadas simplesmente movendo o ícone apropriado para dentro da tela, onde o ícone ficará conectado. A concepção dos ícones é equivalente a escrever um programa ou chamar uma sub-rotina, mas com a vantagem da possibilidade de manipular os dados de forma direta. Não existem operações escondidas, sintaxe ou nomes de comandos para aprender, a filosofia subjacente a estes sistemas é capturada pelo termo WYSIWYG (“*What you see is what you get*”, isto é, “O que

você vê é o que você obtém”). Desta forma, algumas classes de erros de sintaxe são eliminadas. Por exemplo, não é possível mover um objeto que não existe. Utilizando este recurso o sistema necessita de usuários que sejam peritos no domínio das tarefas, mas com o mínimo de conhecimento em computadores e Computação (Hutchins et al., 1986).

A noção de manipulação direta não é a de um conceito unitário, e também nada que se possa quantificar. É uma noção de orientação, que surgiu da necessidade de unir recursos cognitivos adicionais nas interfaces e levou ao sentimento de diretividade. A diretividade é uma impressão ou um sentimento que se tem sobre a interface. Não existem, basicamente, critérios que classifiquem uma interface como sendo mais ou menos direta. A sensação de diretividade é quase sempre relativa, e está relacionada com diversos fatores (Hutchins et al., 1986).

Existem dois aspectos distintos de sentimento de diretividade. Um deles envolve a noção de distância entre o pensamento de uma pessoa e os requisitos físicos do sistema em uso. Uma pequena distância entre os dois significa que a tradução é simples e direta, e os pensamentos da pessoa são rapidamente traduzidos para as ações físicas requeridas pelo sistema e que o sistema está retornando uma resposta de interpretação rápida em termos das metas que interessam ao usuário. Este aspecto é chamado de distância para enfatizar o fato de que a diretividade não é uma propriedade da interface sozinha, mas envolve uma relação entre a tarefa que o usuário tem em mente e a forma como ela está representada na interface. Neste aspecto o que interessa é minimizar o esforço cognitivo necessário para atravessar o golfo entre a meta do usuário e a forma como ela foi representada no sistema. (Hutchins et al., 1986)

O segundo aspecto diz respeito ao grau de sentimento de compromisso, a sensação de estar manipulando o próprio objeto. Existem dois tipos de metáforas no processo de IHC: metáfora conversacional e a metáfora do modelo do mundo. Em um sistema construído com base em uma metáfora conversacional, a interface é uma linguagem intermediária na qual cada um, usuário e sistema, mantêm uma conversa sobre algo assumido, mas não um mundo explicitamente representado. Neste caso, a interface é um meio intermediário entre o usuário e o mundo sobre o qual algumas coisas são ditas. Em um sistema construído sobre a metáfora do modelo do mundo, a interface é um mundo onde o usuário pode agir, e que muda seu estado em resposta as ações do usuário. O mundo de interesse é explicitamente representado e não existem intermediários entre o usuário e o mundo. O uso apropriado da metáfora do mundo real pode criar uma sensação no usuário de estar agindo sobre o próprio objeto no domínio da tarefa. Esta é a chamada diretividade de compromisso (Hutchins et al., 1986).

Tanto a linguagem falada quanto a linguagem de programação dos computadores são simbólicas no sentido de que existe uma relação arbitrária entre a forma do vocabulário e o seu significado. A relação de referência é estabelecida por convenção e deve ser aprendida. Em função da independência entre o significado e a forma, Hutchins, Hollan e Norman descrevem separadamente: Distância Semântica e Distância Articulatória (Hutchins et al., 1986).

A distância semântica reflete a relação entre as intenções do usuário e o significado da expressão na linguagem de interface tanto para entrada, ações que o usuário irá tomar para executar sua intenção, quanto para saída, avaliação do resultado apresentado pelo sistema em relação à intenção do usuário. A distância articulatória reflete a relação entre a forma física da expressão na linguagem de

interface e seu significado, também, para entrada e saída. O objetivo é passar do significado para a forma, na entrada, e da forma para o significado, na saída, com o menor esforço cognitivo possível (figura 4) (Hutchins et al., 1986).

## 2.3 LINGUAGENS VISUAIS

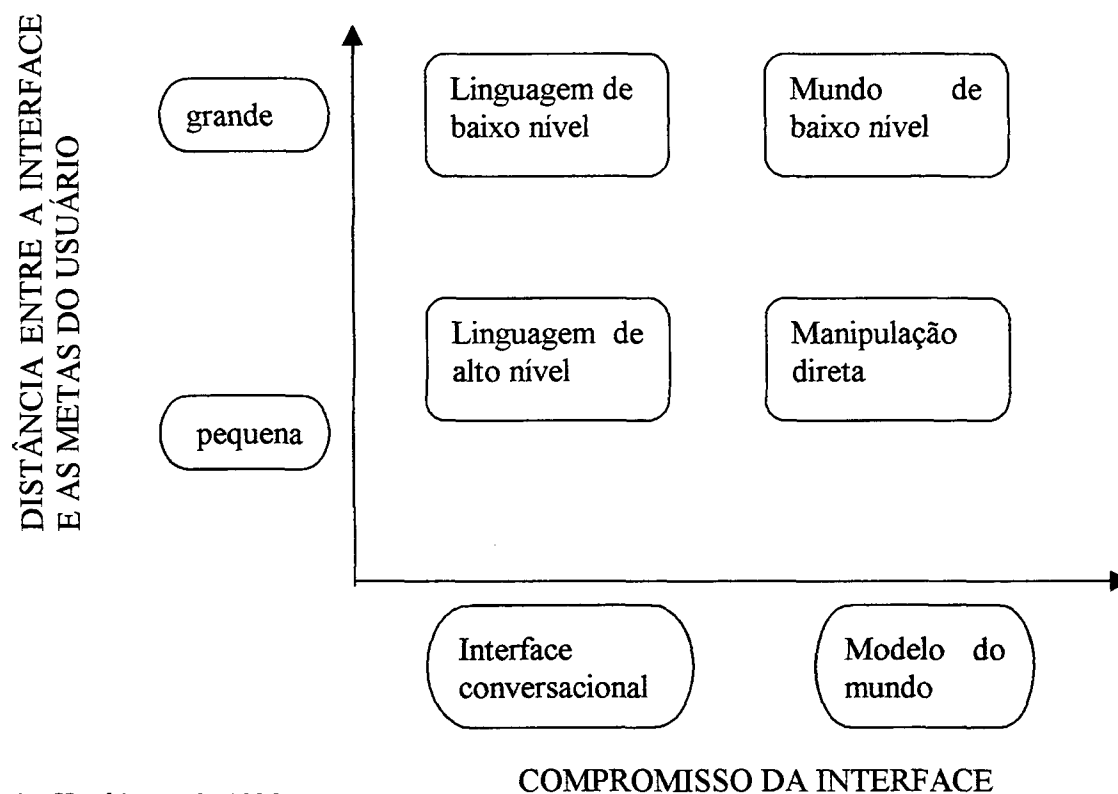
As interfaces podem ser entendidas como linguagens, na medida em que expressam elementos que devem ser usados em um processo de comunicação. Desta forma, a Semiótica é entendida como uma ciência geral de todas as linguagens, suportando formalismos que possibilitam a análise de linguagens de interface humano-computador.

Partindo do princípio de que o principal objetivo da IHC é que o projetista de interfaces se oriente da melhor forma na expressão do modelo funcional do sistema, permitindo ao usuário utilizá-lo plenamente, provendo condições de o usuário entender, aprender e utilizar o sistema de forma criativa, a consistência da linguagem de interface, enquanto um sistema semiótico, é fundamental no processo de comunicação usuário-sistema. Isto porque a consistência da percepção visual do usuário é atingida quando ele pode manipular "intuitivamente" a representação visual. Ela significa a consistência entre o significado pretendido (pelo projetista) e o significado compreendido (pelo usuário). Quando se está tratando de manipulação intuitiva está se fazendo referência às inferências e expectativas de transformação da imagem (forma, estado), que correspondem a conseqüências semânticas operadas pelo sistema em resposta à ação tomada. Se a interface desperta uma certa intenção no usuário, mas bloqueia a manipulação intuitiva, então ocorre uma ruptura de consistência da percepção visual (Martins & De Souza, 1998).

Outro objetivo a ser perseguido por uma boa linguagem de comunicação é que a distância entre os interpretantes de cada agente seja a menor possível. Isto baseado no fato de que a semiótica apresenta uma abordagem específica para o processo de significação, na qual o processo de significação é dinâmico. Nesta abordagem, o processo de significação é apontado como sendo uma tríade formada pelo elemento de conteúdo, elemento de expressão e o interpretante, sendo o signo um produto dinâmico desta relação na presença de um intérprete. Assim o interpretante é aquilo que o signo produz na mente do intérprete e pode ser concebido como uma outra representação referida ao mesmo objeto, uma vez que cada agente tem seu próprio interpretante sobre o que está sendo comunicado. Isto posto, quanto mais próximo o interpretante do agente receptor estiver do interpretante do agente emissor, maior será a chance de que a mensagem seja compreendida. O processo de comunicação é uma negociação constante entre os interpretantes dos agentes. Uma vez que o signo visual percebido pelo usuário pode não ser o mesmo para outro usuário e não ser o mesmo do projetista, a Linguagem Visual (LV) vem buscar uma aproximação, de forma que o signo do projetista seja reconhecido por um grande número de usuários, no maior número possível de circunstâncias. Para tanto o projetista deve definir a LV antes de especificar a interface. A definição prévia permite estruturar o meio de expressão com características mínimas de código, correlacionando elementos da expressão e do conteúdo de forma sistemática. Na medida em que esta correlação incorpore um padrão cognitivo, o signo visual do projetista tem melhores chances de ser reconhecido pelo usuário e este pode expressar melhor suas intenções, atingindo os objetivos de usabilidade propostos (figura 4) (Martins & De Souza, 1998).

Todo este processo permite uma consistência entre o significado pretendido e o significado percebido, que ocorre quando o projetista, através da LV, expressa adequadamente o modelo funcional do sistema.

Figura 4: Espaço da Interface definindo as categorias de linguagens



Fonte: Hutchins et al., 1986

Uma linguagem visual para tratamento de base de dados não deve ter somente uma interface apropriada para dar suporte para a geração da consulta de forma gráfica, mas deve ser capaz de proporcionar ao usuário um conjunto de estratégias metodológicas para diminuir o esforço cognitivo necessário durante a interação com o sistema. Isto se deve ao fato de que o usuário necessita não só de uma linguagem para a criação da consulta, mas de um meio de comunicação que seja capaz de diminuir a distância semântica gerada por diversas dificuldades que

surtem enquanto o usuário está trabalhando em um domínio desconhecido, tais como: descobrir informações implícitas, analisar esquemas muito detalhados, necessidade de armazenar informações pré-requeridas, entre outras. Desta forma, a idéia é oferecer uma ferramenta que apresente um conjunto de primitivas gráficas para auxiliar o usuário no processo de entendimento do esquema da base de dados e que facilite ao máximo a extração das informações relevantes (Angelaccio et al., 1990 (b)).

Conforme a metodologia apresentada por De Souza em (De Souza et al., 1999 (a)), as características e classes dos usuários que irão fazer uso do sistema e as tarefas que tais usuários irão executar devem ser analisadas com clareza e entendidas pelo projetista do sistema. Assim sendo, o sistema proposto deve ser construído para facilitar a atuação do usuário no que diz respeito à realização das tarefas que ele tem em mente. Também, o uso de signos e ícones para representar a realidade do usuário deve levar em conta as análises de metáforas feitas por Baranauskas (Oliveira & Baranauskas, 1999) (Baranauskas et al., 1998). Quando se faz uso de signos, ícones e representações gráficas em geral, está sendo utilizada uma linguagem visual que deve seguir um formalismo visual para que o usuário tenha bem definidos as regras de formação desta linguagem, mesmo que sendo realizada por meio de gráficos. Esta necessidade fica bem clara nos estudos de Martins (Martins & De Souza, 1998).

## 2.4 DESIGN DE INTERFACES HUMANO-COMPUTADOR

No que diz respeito à Engenharia Semiótica, o que o designer transmite por meio da interface não é uma mensagem como a de um documento (livro ou filme),



mas uma mensagem interativa e dinâmica: um sistema de comunicação (para a interação) e um resolvidor de problemas (a funcionalidade da aplicação).

Visto por esta perspectiva, o design de interface envolve não apenas a concepção do modelo de interação, mas a comunicação deste modelo de maneira a revelar para o usuário o espectro de usabilidade da aplicação. A usabilidade é o conceito que avalia a qualidade da interação de sistemas computacionais interativos com os seus usuários. Ela procura determinar em que medida o software pode ser usado por usuários específicos para alcançar metas específicas como efetividade, eficiência e satisfação em um contexto específico de uso (Prado & Baranauskas, 2000). Seu maior desafio é conseguir que o usuário adquira a competência necessária para tirar proveito de todo o potencial de uma aplicação que tenha sido projetada como uma ferramenta intelectual. Num processo simbiótico, a ferramenta intelectual visa potencializar a usabilidade aumentando a competência do usuário (Leite, 1998). O aspecto de usabilidade que a ES visa resolver é como o conhecimento que o usuário precisa adquirir para utilizar melhor o sistema pode ser melhor “ensinado” através da interface de usuário (Leite & De Souza, 1999).

A partir deste contexto é possível admitir que o modelo de funcionalidade determina o que o usuário pode fazer enquanto o modelo de interação determina como ele pode utilizar o sistema, o conjunto destes dois modelos é chamado de modelo de usabilidade por Leite em (Leite, 1998). Manuais e auxílios são outra forma usada pelo projetista para comunicar este modelo; no entanto a comunicação ocorre da forma direta (primariamente escrita) e externa à interface do sistema em si – mesmo sendo em uma janela de auxílio, trata-se de outra janela (Prado & Baranauskas, 2000).

Projetar interfaces é, portanto, projetar uma mensagem complexa, interativa e unidirecional, destinada a usuários de aplicações computacionais. Diversos elementos da interface podem possuir significados distintos para o designer e para o usuário. Botões, palavras, cores, menus, entre outros, quase tudo na interface tem o potencial de ser signo. O designer necessita controlar este processo de comunicação para melhorar a usabilidade do sistema. Ele precisa projetar a interface consciente de que está projetando um signo cuja expressão é formada por outros signos que devem ativar interpretantes que conduzam ao modelo de usabilidade (Leite, 1998).

Trabalhando desta maneira o designer tem condições de atingir com maior probabilidade de acerto o objetivo de comunicar ao usuário duas perguntas fundamentais da ES (De Souza, 1993): “qual a interpretação do designer sobre o(s) problema(s) do usuário?” e “como o usuário pode interagir com a aplicação para resolver este(s) problema(s)?” Onde o usuário irá conceber a resposta a estas perguntas à medida que interage com a aplicação.

As habilidades do usuário, a situação de uso e o contexto onde ele está envolvido (o domínio da aplicação) são fundamentais no processo de interação e devem ser considerados no design das aplicações de software (Leite, 1998).

O conhecimento que o usuário demonstra sobre a tarefa a ser desenvolvida em forma computacional é denominado por Norman de modelo conceitual do usuário (Norman, 1986). Desta forma, a competência do usuário é o conhecimento que ele tem a respeito da interface e sua funcionalidade que são necessários para ele desempenhar a interação com o sistema, isto é, o modelo de interface e da funcionalidade que o usuário deve adquirir. Como este conhecimento é um fator

relevante para a usabilidade, Leite em (Leite, 1998) denominou-o de modelo de usabilidade da aplicação.

Norman considera ainda em (Norman, 1986), que a maneira de se adquirir o *modelo conceitual do usuário* é por meio da interface e da documentação que acompanha o sistema, às quais Norman (Norman, 1986) denomina de imagem do sistema. A imagem do sistema deve veicular o modelo de design da maneira mais clara e óbvia possível para que os usuários não desenvolvam um modelo conceitual incorreto. Assim, tudo no sistema deve estar direcionado a tornar este modelo coerente e lógico, uma vez que ele determina as atividades cognitivas que o usuário deverá desempenhar para melhor utilizar o sistema.

A interface de usuário é a parte do artefato de software com a qual o usuário entra em contato de forma física, perceptiva e cognitiva na realização de tarefas no seu domínio de atividades. A interface de usuário é composta por uma coleção de dispositivos por meio dos quais o usuário pode trocar informações com o sistema. Esta troca ocorre através de estruturas de interação tais como menus, janelas, ícones, linguagens de comando, formulários, perguntas e respostas em linguagem natural, dentre outras que determinam o sistema de interação (Leite, 1998).

O design de interface de usuário é considerado o processo de concepção do sistema de interação. E a interface pode desempenhar um papel importantíssimo na aquisição do modelo de usabilidade pelo usuário, uma vez que ela deve ser a ferramenta de comunicação que veicula e ensina o modelo de usabilidade (o modelo do design) para o usuário (Leite, 1998).

Desta forma, segundo Leite em (Leite, 1998), a usabilidade de um sistema depende de dois aspectos fundamentais: a) o design da interação e da funcionalidade da aplicação e b) a aquisição deste modelo de usabilidade pelo

usuário. A funcionalidade deve estar adequada às tarefas dos usuários, inserida e motivada pelo domínio da aplicação, e deve poder ser entendida pelo próprio usuário. A idéia principal de Leite em (Leite, 1998) é que o modelo de usabilidade pode ser adquirido durante o processo de interação pela interpretação das mensagens da interface de usuário codificadas na Linguagem de Design de Interface de Usuário – LDIU.

É importante conhecer o conceito de funcionalidade, definido por Leite em (Leite, 1998)

*O termo funcionalidade designa a parte conceitual do sistema computacional que contém as funções necessárias para a resolução de problemas em um domínio específico. A funcionalidade se refere à que um programa faz e, no caso de software interativo, o que ele deve oferecer para seus usuários. Funcionalidade é diferente de funcionamento que se refere à forma como um software realiza suas funções.*

A interface, como produto a ser desenvolvido, não é uma máquina, mas um conjunto de mecanismos físicos e lógicos para controle e comunicação de uma aplicação de software. Os mecanismos físicos são os dispositivos de entrada e saída, tais como teclado, botões, monitor de vídeo, entre outros. Os dispositivos lógicos, conceituais ou virtuais são os signos e linguagens que permitem o processo de comunicação entre o usuário e o sistema.

#### 2.4.1 Auxílio ao Usuário

Um bom design do sistema e da imagem do sistema tem como finalidade a redução do número de problemas que o usuário irá enfrentar no uso do mesmo. No entanto, é importante que o processo de design busque também proporcionar informações para que os usuários possam aprender a usar o sistema de maneira eficiente, corrigir os seus erros e achar outros tipos de funcionalidade que estejam

disponíveis. Os auxílios *online* freqüentemente encontrados em ambientes de interface proporcionam um ou mais tipos de auxílio como por exemplo: mensagens de ajuda para seleção de um objeto da interface; auxílio sensível ao contexto na forma de diagramas de estado ou caixas de diálogo; textos de ajuda genéricos; telas de auxílio adicional assim como, extensa documentação disponível em papel ou *online* (Preece et al., 1994).

Os auxílios devem procurar responder a algumas perguntas típicas dos usuários que podem se concentrar em algumas interjeições que refletem problemas de ruptura de comunicabilidade. Estes problemas de comunicabilidade podem ser classificados conforme De Souza em (De Souza et al. 1999 (b)) e esta classificação foi aproveitada aqui para estabelecer parâmetros para a construção do auxílio *online*, que são (García, 2000): exploração de meta (“o que eu posso fazer com este sistema?”); definição e descrição (“o que é isto? Para que serve isto?”); realização de tarefas (“como posso fazer isto?”); diagnóstico (“como foi acontecer isto?”) e identificação de estado (“onde estou?”).

Os tipos de modelos cognitivos de apoio à confecção de material de auxílio são os modelos estrutural e funcional. O modelo estrutural descreve a estrutura do sistema e busca mostrar como o sistema funciona. O modelo funcional ou procedural fornece informação procedural sobre como o sistema deve ser usado, isto é mostra como usar (Preece et al., 1994).

O modelo estrutural explica como o sistema funciona fazendo com que o usuário consiga fazer previsões acerca de como o sistema irá se comportar, o que permite ao usuário descobrir como realizar certas tarefas no sistema, aprendendo com ele. Em uma interface o modelo estrutural corresponde à descrição do ambiente de interface, com suas partes constituintes (Preece et al., 1994).

Os modelos funcionais também chamados de procedurais ou *task-action mapping models*, fazem a associação entre o ambiente real (do domínio da aplicação) e o ambiente da ação ou computacional (o sistema). É importante que os designers deixem aparente a relação entre esses dois níveis uma vez que este modelo irá se apropriar do conhecimento anterior do domínio da aplicação e de domínios de tarefas semelhantes que o usuário vem desenvolvendo no mundo real. Esta dependência do domínio do modelo funcional torna-o dependente do contexto, deixando-o mais fácil de usar e de integrar a outros modelos previamente adquiridos, enquanto os modelos estruturais são livres de contexto (Preece et al., 1994).

Partindo do fato de que um modelo mental de um sistema consiste de elementos e relacionamentos entre eles, o auxílio funcional deve abordar funções disponíveis no ambiente (expressas da forma como o usuário enxerga-as no mundo real), descrevendo as ações (computacionais) que devem ser realizadas pelo usuário para que a função seja executada pelo sistema (Preece et al. 1994).

Para confeccionar o auxílio ao usuário este deve necessariamente ter a abordagem funcional e pode, adicionalmente, incorporar a abordagem estrutural. Um auxílio completo consiste de uma parte da função para a aplicação (abordagem funcional) e outra da aplicação para a função (abordagem estrutural). É importante também a utilização de linguagem do domínio da aplicação que seja familiar ao usuário nos itens e textos descritivos do auxílio *online* (Preece et al., 1994).

## 2.5 AVALIAÇÃO DE INTERFACES HUMANO-COMPUTADOR

Duas das maiores dificuldades da área de IHC consistem em encontrar um modelo de design que chegue o mais perto possível do modelo do usuário e em traduzir este modelo de design para a imagem do sistema, implicando um menor

esforço mental por parte do usuário para conseguir atravessar os golfos de execução e avaliação (Norman, 1986). Com o objetivo de quantificar quão próximo o modelo de design se encontra do modelo do usuário, é necessário utilizar alguma metodologia de avaliação. Duas grandes vertentes teóricas complementares podem ser aproveitadas na avaliação de interfaces. Sob a ótica das Ciências Cognitivas, as métricas cognitivas, utilizadas conjuntamente com o estereótipo do usuário, permitem atribuir graus de dificuldade ao uso de ferramentas computacionais (Pimentel & Direne, 1998) e (Pimentel & Direne, 1997); sob o olhar da Engenharia Semiótica (De Souza, 1993) e (De Souza, 1996), as linguagens de interface podem ser medidas em relação à sua capacidade de transmitir ao usuário qual o modelo da aplicação concebido pelo designer.

As abordagens cognitivas para o projeto de interfaces normalmente buscam modelos do funcionamento da mente humana tentando entender quais características as interfaces devem ter para que elas sejam mais fáceis de usar e de aprender. Estas abordagens são de fundamental importância para a elaboração do modelo conceitual da aplicação. Elas, entretanto, deixam de considerar o amplo potencial comunicativo que as interfaces têm e que pode ser explorado no aprendizado deste modelo. A ES tem como objetivo explorar esta característica de meta-comunicação que os sistemas computacionais possuem e oferecer ao projetista instrumentação que lhe permita ensinar quais soluções ele projetou para os problemas dos usuários. O conceito de signo, de interpretante e de semióse ilimitada, abordados pela ES, revelam o importante papel que os sistemas semióticos desempenham na aquisição de conhecimento e a limitação que os modelos cognitivos possuem por não abordar tais fenômenos como processos semióticos (Prado & Baranauskas, 2000).

Métodos e técnicas de avaliação de interfaces devem ser considerados no contexto do modelo de projeto do software; subjacente a esse contexto existe um entendimento para o conceito de interface que orienta objetivos de avaliação (Prado & Baranauskas, 2000).

Tradicionalmente, os métodos de avaliação de interfaces têm usado como referencial o conceito de usabilidade. Entretanto, a maioria destas abordagens apresenta como solução a este desafio modelos e técnicas para o projeto pautadas em características cognitivas dos usuários (Norman, 1986). Segundo Leite em (Leite & De Souza, 1999), estes modelos cognitivos são insuficientes para o desafio de usabilidade, uma vez que, conforme Oliveira e Baranauskas em (Oliveira & Baranauskas, 1998 (a)), não levam em consideração que o software deve existir dentro de um contexto de valores e necessidades das pessoas que irão utilizá-lo, e neste ponto devem ser avaliados também no sentido de articularem o papel do projetista e da interface na aquisição do modelo de usabilidade pelo usuário. Desta forma a ES aplicada ao design de software vem preencher esta lacuna, na qual sistemas são considerados como artefatos de meta-comunicação (De Souza, 1993).

Diversas metodologias de avaliação de interfaces têm sido propostas utilizando a Semiótica como referencial teórico para criar notação e procedimentos que possibilitem uma análise objetiva de linguagens de interface de software. Dentre elas pode ser citada a Linguagem de Especificação da Mensagem do Projetista (LEMD) proposta por Leite em (Leite, 1998) que tem por objetivo específico apresentar uma mensagem do designer que comunica para os usuários o modelo de interação e de funcionalidade do sistema, o modelo de usabilidade. Proposta inicialmente como auxílio ao design de interfaces, esta linguagem foi posteriormente



utilizada por Prado e Baranauskas (Prado & Baranauskas, 2000), num processo de avaliação.

### 2.5.1 Linguagem de Especificação da Mensagem do Projetista

A contribuição da LEMD para a ES é a possibilidade de uma especificação da interface em nível abstrato, que permite a representação de associações entre intenções de comunicação do projetista e expressões de interface utilizadas para este fim, como, por exemplo, os *widgets* presentes nas ferramentas de GUI (*Graphical Users Interface*), detalhados em (De Souza et al., 1999 (a)).

A principal vantagem de se utilizar a LEMD consiste em que a medida em que ela reflete um modelo global de comunicação humana por meio de artefatos de meta-comunicação (as interfaces de sistema) estimula o projetista de interface a considerar não somente os processos interpretativos do usuário, mas principalmente os processos comunicativos e expressivos que ele pode utilizar com maior consciência e intencionalidade para atingir metas comuns a todos os que visam aumentar a usabilidade de aplicações computacionais (Leite & De Souza, 1999).

A partir da LEMD é possível declarar os signos do domínio da aplicação que estarão presentes na interface para visualização e manipulação pelo usuário, especificar estruturas de interação que correspondam ao modelo de interação planejado pelo projetista, e mesmo representar seqüências completas de passos que o usuário deve executar para atingir uma determinada meta do seu escopo de trabalho (Prado & Baranauskas, 2000). Desta forma a LEMD permite estruturar os diversos tipos de mensagens que o projetista necessita enviar para comunicar para o usuário o que ele pode fazer (as funções de aplicação que operam sobre os signos do domínio) e como ele pode utilizar o sistema (os comandos de função). Além

disso, a LEMD possibilita ao projetista especificar as diversas outras mensagens que auxiliam o processo de utilização e aprendizado do sistema (Leite & De Souza, 1999).

A LEMD diferencia diversos tipos de mensagens, conforme (Leite & De Souza, 1999):

- Mensagens sobre estados de signos do domínio: revelam o estado do sistema e permitem ao usuário avaliar se sua meta foi atingida (*View information-of <domain-sign>*);
- Mensagens sobre funções da aplicação: revelam o estado das funções disponibilizadas pelo sistema (*View State-of <application- function>*) e o que o usuário deve fazer para controlá-las;
- Mensagens sobre a estrutura sintática dos comandos: revelam a estrutura da interação que o usuário precisa desempenhar. A estrutura sintática determina como as interações básicas podem ser articuladas na formação de comandos compostos. As interações podem ser agrupadas em seqüências (*sequence*), repetição (*repeat*), agrupamento (*join*), combinação (*combine*) e seleção (*select*);
- Mensagens sobre interações básicas: indicam ao usuário a interação a ser desempenhada. As interações básicas previstas pela linguagem são acionar (*activate*), fornecer informação (*enter*) e selecionar informação (*select*);
- Mensagens de meta-comunicação de assistência a tarefas: auxiliam o usuário a realizar tarefas compostas por mais de um comando, apresentando a tarefa decomposta em passos (*Task-Message*);

- Mensagens de meta-comunicação para apresentação e controle da leitura da mensagem: comunicam como o usuário deve ler a própria mensagem do projetista. Corresponde à estrutura de navegação entre telas (*Show Command-message*);
- Mensagens de meta-comunicação direta: permitem ao projetista enviar uma mensagem diretamente ao usuário para se referir a qualquer outro elemento da interface. Estas mensagens são especificadas através do elemento *View* da linguagem.

Um exemplo simples da LEMD, extraído de Prado e Baranauskas em (Prado & Baranauskas, 2000), pode ser o design de interface de sistemas para consulta de saldos num terminal bancário de auto-atendimento. A estrutura de interação pode ser representada da seguinte forma:

```
Sequence {
    View "insira o cartão"
    Enter information-of senha
    Show information-of saldo
}
```

No exemplo acima, pode ser observado que no comando *show* a representação do conteúdo da mensagem é feito de forma independente de sua expressão, isto é, de como ela será apresentada na interface do sistema. Para realizar essa transposição, o designer necessita de regras de mapeamento semântico que associem cada instrução da LEMD a um elemento de interface de acordo com o significado que este é capaz de transmitir ao usuário. Exemplos destas regras, extraídos de Leite (Leite & De Souza, 1999), são mostrados na tabela 1:

Tabela 1: Exemplos de regras de mapeamento semântico

Elemento da LEMD	Widget associado	Significado
View	Labels ou <i>message-boxes</i>	Mensagem direta do designer para o usuário
Activate	Botões de pressão	Acionamento de uma função da aplicação
Select information-of	<i>Radio buttons</i> ou <i>combo boxes</i>	Escolher uma opção de uma lista
Join	Painéis ou <i>boxes</i>	Agrupamento

Fonte: Leite & De Souza, 1999

Segundo estudos de Leite (Leite & De Souza, 1999) e Prado e Baranauskas (Prado & Baranauskas, 2000), a LEMD permite analisar as intenções comunicativas do projetista e pode ser a base para a avaliação de interfaces existentes (Leite & De Souza, 1999). Esta avaliação é motivada pela necessidade de localizar da forma mais precisa possível, partes da interface onde as mensagens do projetista não refletem o modelo de usabilidade (Prado & Baranauskas, 2000).

Conforme Prado e Baranauskas (Prado & Baranauskas, 2000), o procedimento de avaliação consiste em o avaliador assumir a posição de receptor da mensagem, decodificando-a e documentando-a na forma de LEMD sem qualquer expectativa sobre a mensagem que deveria encontrar. O avaliador deve, portanto, captar o modelo de usabilidade expresso na interface pela mensagem do projetista, verificar o grau de dificuldade na sua detecção e compará-lo ao modelo intencionado.

O processo de decodificação da interface pelo avaliador proposto em (Prado & Baranauskas, 2000) leva em consideração o perfil do usuário-alvo, em especial do seu grau de familiaridade com o código que constitui o padrão de *widgets* usado na construção da interface em avaliação. A “tradução” da representação visual da

interface para sua descrição correspondente em LEMD é feita segundo a semântica dos *widgets*, do seu leiaute e da própria dinâmica dos elementos. Durante o processo de interpretação da interface e escrita da LEMD, o avaliador destaca pontos onde a mensagem recebida é deficiente ou dúbia, ou seja, onde um usuário com o perfil pretendido não compreenderia a intenção do projetista expressa através do código visual (*widgets* e leiaute). Estes pontos são assinalados na escrita da LEMD grifando-se os trechos onde, na visão do avaliador, a comunicação do significado da meta-mensagem é incerta. Eles sinalizam uma possível lacuna na comunicação projetista-usuário pois o avaliador, ao situar-se em termos das capacidades interpretativas do perfil do usuário pretendido para o sistema, não é capaz de definir exatamente o significado que o usuário conseguiria perceber.

A partir desta documentação em LEMD do que o avaliador considera ser o modelo de usabilidade mais provável de ser apreendido pelo usuário, é necessário determinar o quão próximo o modelo de usabilidade captado está das intenções do projetista. Para isso, é preciso ter acesso a outras fontes do modelo de usabilidade, para comparação. Estas fontes devem ter um grau de indireção menor que o da meta-comunicação via interface, sendo ideal o contato direto com o próprio projetista. Na impossibilidade desse contato direto, documentação *online* e manuais de usuário também são fontes válidas (Prado & Baranauskas, 2000).

Do confronto entre modelo de usabilidade descrito em LEMD (meta-linguagem) e as fontes alternativas, pode-se detectar deficiências na comunicação projetista-usuário, onde, por exemplo, o usuário entende que deve utilizar um modo de interação com o sistema diferente do que o projetista intencionara. Tais pontos da interface sugerem que o usuário não estaria recebendo convenientemente a mensagem do projetista (Prado & Baranauskas, 2000).

### 2.5.2 Escolha do Método Semiótico de Análise

A execução da avaliação proposta por Prado e Baranauskas, utilizando a LEMD em duas etapas (interpretação da interface e documentação da mensagem do projetista, e posterior comparação com o modelo de usabilidade) mostra-se interessante por minimizar a predisposição do avaliador na interpretação da mensagem do projetista presente na interface. Na tentativa de confrontação imediata do modelo de usabilidade dos manuais e auxílios com o da mensagem da interface, o avaliador é levado a localizar na interface as informações sobre o modelo de usabilidade que foram coletados no auxílio. Esse procedimento não permite a desejada “neutralidade” do avaliador como alguém que está recebendo tais informações de usabilidade a partir da mensagem de meta-comunicação do projetista (Prado & Baranauskas, 2000).

Outros métodos foram analisados, inclusive alguns com o benefício de serem mais exatos no sentido de não dependerem da interpretação subjetiva dada por um avaliador, como a análise glossemática em (Oliveira & Baranauskas, 1998 (b)). Na sua maioria os métodos estudados dispensam a presença do usuário, o que facilita e torna mais barata a análise das interfaces escolhidas. Porém o método LEMD permite uma análise mais abrangente de todos os meios de comunicação entre projetista-usuário, isto é, a análise é realizada levando em conta os signos e as mensagens enviadas ao usuário, além dos auxílios e assistentes. Uma segunda vantagem é que o uso da LEMD na análise permite que os registros de falhas na comunicação possam ser corrigidos na proposta da nova linguagem de interface. Outro benefício do uso da LEMD é a possibilidade de utilizar este mesmo formalismo para a especificação da linguagem visual de interface, objetivo deste trabalho.

Desta forma, a idéia aqui é utilizar, na avaliação da ferramenta de consulta de hipótese, o formalismo da ES para verificação da efetividade da comunicação projetista-usuário presente na interface, baseada no trabalho de Prado e Baranauskas (Prado & Baranauskas, 2000), que propõem uma avaliação de interface fazendo uso dos conceitos de Linguagem de Especificação da Mensagem do Projetista (LEMD) propostos por Leite e De Souza em (Leite & De Souza, 1999).

### 3. METODOLOGIA PARA CONSTRUÇÃO DA INTERFACE

Como visto no capítulo anterior a prática de design de interfaces é conduzida baseada em princípios, diretrizes ou padrões e que são necessárias teorias que apoiem este processo, melhorando a usabilidade do sistema. Entretanto, pode-se perceber que por ser um campo de estudo recente que agrupa pesquisadores de diversas áreas, a IHC não apresenta teorias consagradas para o design, desenvolvimento e avaliação de interfaces de usuário. O que foi possível identificar foram diferentes propostas elaboradas por pesquisadores que aplicam conhecimento proveniente de disciplinas específicas nas diversas etapas do design. Como a ES apresenta uma nova perspectiva que reformula e busca apresentar soluções teóricas ela é uma abordagem e deve apresentar um esquema para a aplicação destas teorias no processo de design. Assim, o objetivo deste capítulo é descrever as etapas desta abordagem que será aplicada no desenvolvimento do design da interface proposta neste trabalho.

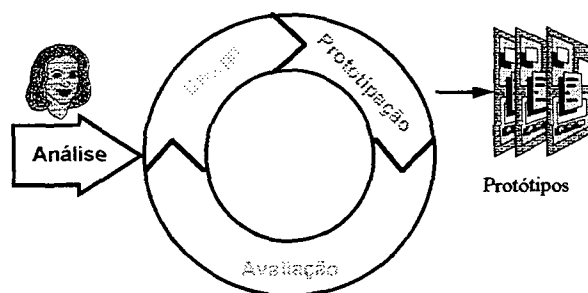
#### 3.1 DESCRIÇÃO DA METODOLOGIA

Para a construção da interface proposta para a seleção do sub-esquema de consulta baseado na ferramenta VIQUEN, foi utilizada a abordagem da ES, proposta por De Souza em (De Souza et al., 1999 (a)). Nesta abordagem, o desenvolvimento da interface de usuário ocorre num processo formado por quatro etapas básicas: análise, design, prototipação e avaliação. Estas três últimas atividades ocorrem num processo cíclico no qual o design da interface pode ser corrigido ou evoluir para novas etapas a partir da avaliação da interface, como mostra a figura 5. O design é



visto como um processo de produção de mensagem e deve ser conduzido em duas fases: a formulação do conteúdo e a elaboração da expressão da mensagem do designer para o usuário. Os métodos utilizados para a realização das duas etapas citadas serão descritos na seqüência.

Figura 5: O processo de design de interfaces na Engenharia Semiótica



Fonte: Leite, 1998

Sendo a interface responsável por fazer o usuário ter condições de interagir com a funcionalidade do sistema, fica claro que o design da interface de usuário depende da especificação dos modelos de interação e da funcionalidade do sistema (De Souza et al., 1999 (a)). A especificação da funcionalidade visa descrever quais funções o sistema deve oferecer para os usuários. A especificação do modelo de intenção visa descrever de forma abstrata e precisa como o usuário pode interagir com o sistema independentemente de quais dispositivos de interação ou *widgets* ele vai utilizar e de como este processo de interação será implementado pelo software da interface. A partir desta descrição o designer da interface deve escolher qual o modelo de interação específico (estilo ou padrão) o usuário irá utilizar para interagir. A descrição abstrata oferece a vantagem de ser independente de estilo e ferramenta de interface (De Souza et al., 1999 (a)).

No desenvolvimento de sistemas centrado no usuário a especificação da funcionalidade e a do modelo de interação são derivadas do modelo de tarefas, que deve considerar os diferentes perfis de usuários, e são a base para o restante do desenvolvimento (De Souza et al., 1999 (a)).

### 3.1.1 Análise de Usuários e Tarefas

O usuário deve ser sempre o foco central de interesse do projetista ao longo do design da interface. Logo, é necessário identificar quem são os usuários e caracterizá-los, isto é, especificar quais funções exercem, quais capacidades possuem, entre outros. Além disto, também é necessário realizar a análise de tarefas, cujo objetivo consiste em fornecer ao designer a visão que os usuários têm das tarefas que eles precisam realizar (De Souza et al., 1999 (a)).

### 3.1.2 Mapeamento Tarefa-Ação

As abordagens cognitivas caracterizam-se principalmente por considerar o processo de interação como um problema de mapear (associar) o conhecimento que se tem a respeito da funcionalidade e da interface do usuário do sistema (modelo conceitual que o usuário tem do sistema) com o modelo que ele tem da tarefa (o modelo conceitual da tarefa). Os problemas identificados nesta perspectiva envolvem o estudo das atividades mentais necessárias à interação, como a recordação de comando e regras e a interpretação de símbolos e resultados. Este mapeamento apenas pode ocorrer no momento em que o usuário tiver adquirido o referido modelo conceitual do sistema. De acordo com Norman em (Norman, 1986) a abordagem cognitiva considera este problema dividido em dois sub-problemas: o

mapeamento tarefa-ação que o usuário precisa desempenhar (a travessia do golfo de execução); e o problema da avaliação (a travessia do golfo de avaliação).

O mapeamento tarefa-ação abrange o conhecimento que o usuário precisa ter para poder realizá-lo e inclui modelos de tarefa, estruturação de plano e especificação da seqüência de ações, conforme o modelo apresentado por Norman em (Norman, 1986). Este mapeamento pode ser estudado por meio das distâncias semântica e articulatória, conforme Hutchins em (Hutchins et al., 1986). A distância semântica avalia a separação entre as metas/tarefas do usuário e a funcionalidade do sistema associada a elas, isto é, a existência de um comando no modelo de interação cujo significado (execução) seja aquele pretendido pelo usuário. Uma distância pequena implica existir um comando associado à meta, enquanto que uma distância grande indica que o usuário precisa quebrar metas em submetas e realizar um planejamento de tarefas. A distância articulatória avalia o relacionamento entre a forma da seqüência de ações (o comando) e seu significado.

### 3.1.3 Modelagem de Tarefas

A modelagem de tarefas consiste em formalizá-las de forma a mapeá-las na interface gráfica. Existem diversos modelos de especificação de tarefas, cada qual com notação e objetivos específicos, dentre os quais podem ser destacados a TAG (*Task-Action Grammar* – Gramática Tarefa-Ação) (Payne & Green, 1986), UAN (*User Action Notation* – Notação da Ação do Usuário) (Hix & Hartson, 1993), GOMS (*Goals, Operators, Methods and Selection Rules* – Metas, Operadores, Métodos e Regras de Seleção) (Card et al., 1983).

### 3.1.3.1 *Task-Action Grammar* – Gramática Tarefa-Ação

A TAG é uma gramática orientada a tarefas e faz parte do grupo de gramáticas de ação que são modelos gramaticais para o mapeamento entre tarefas dos usuários. Os autores em (Payne & Green, 1986) argumentam que estas gramáticas são representações mentais da linguagem de interação e do modelo de interação. Estes modelos de gramáticas de ação visam determinar um modelo de interação cujas categorias reflitam as categorias do modelo semântico. A idéia básica é de isomorfismo entre os modelos conceituais e de tarefas, para que possam ser mapeados diretamente na semântica e sintaxe do modelo de interação.

A TAG, um dos formalismos mais conhecidos para modelos de gramáticas de ação é uma gramática gerativa, onde são elaboradas linguagens de comando a partir de modelos de tarefas. A técnica consiste em identificar tarefas simples, representá-las de forma categorizada e por meio de regras de produção. Esta notação permite tratar diversos tipos de consistência, nos eixos lexical, sintático, semântico e sintático-semântico, e permite, ainda, verificar a completude do modelo (Payne & Green, 1986).

A TAG captura consistência pela representação da competência de um usuário especialista por meio de uma notação formal. O objetivo é facilitar o aprendizado e melhorar o desempenho do usuário. Assim, a ênfase está em resolver os problemas relativos à consistência. Logo a TAG é uma meta-linguagem e um modelo de representação mental de linguagens de tarefas (Payne & Green, 1986). Esta gramática é geralmente utilizada para tratamento de interfaces textuais, desfrutando de reconhecimento e aceitação bastante grandes no meio acadêmico.

### 3.1.3.2 *User Action Notation* – Notação da Ação do Usuário

A UAN é utilizada principalmente para interfaces de manipulação direta, onde representa aspectos do comportamento do sistema do ponto de vista do usuário, ou seja, que tarefas e ações o usuário realiza na interface. Este modelo associa as ações do usuário ao *feedback* do sistema, ao estado do sistema e ao modelo computacional da aplicação (Hix & Hartson, 1993).

### 3.1.3.3 *Goals, Operators, Methods and Selection Rules* – Metas, Operadores, Métodos e Regras de Seleção

O modelo GOMS pretende representar o comportamento dinâmico da interação com o computador, com base num modelo do comportamento humano que possui 3 subsistemas de interação: o perceptual (auditivo e visual), o motor (movimentos braço-mão-dedo), e o cognitivo (tomada de decisão e acesso à memória). Estes comportamentos são definidos por metas, operadores, métodos e regras de seleção (Card et al., 1983).

### 3.1.3.4 Escolha do método de modelagem de tarefas

Abordaremos a análise de tarefas por meio do modelo TAG, uma vez que um de seus objetivos é o de refletir a estrutura de raciocínio do usuário, em função da tarefa (Martins & De Souza, 1998). Isto é possível por meio da inclusão de elementos que parametrizam as regras da gramática com o objetivo de enfatizar a consistência e codificar o mundo real do usuário, por exemplo, “para cima” é o oposto de “para baixo”. A TAG dispõe de ferramentas para representar o “conhecimento do mundo real”, fazendo uso de uma forma especial (*known-item*) que é utilizada para denotar informações que o usuário já conhece, e desta forma a

necessidade de aprendizado, por parte do usuário, para utilizar o sistema é minimizada (Payne & Green, 1986). Porém, o designer é responsável pela inserção deste conhecimento de mundo na descrição da TAG e sua validade dependerá do seu julgamento profissional.

É importante salientar que na MD a regularidade da expressão do usuário, dentro do aplicativo e entre aplicativos, é importante porque a forma de ativação das ações não está ligada a nenhum elemento particular da interface, e, portanto, o usuário fará a tradução articulatória de forma intuitiva. Por exemplo, existe um modelo de interação associado à idéia de botão, no qual o botão deve ser clicado, logo a presença visual do botão remete à idéia de ser pressionado. No caso do traçado de linhas não existe nenhum elemento visual sugestivo. O único elemento sugestivo é metafórico, ligando a idéia ao uso do mouse como extensão da mão. Portanto neste caso a regularidade de expressão, por meio do tipo expressivo adequado, é um elemento de reforço da metáfora, logo, é um forte elemento cognitivo.

A TAG será utilizada no âmbito da *Semiotic TAG* (STAG) (Martins, 1998) que é uma notação para auxílio à especificação de linguagens visuais por manipulação direta, que orienta a sistematização de escolhas expressivas de acordo com os princípios semióticos, sem exigir a competência de um semioticista. Ela foi proposta como uma extensão da TAG.

Segundo Martins em (Martins & De Souza, 1998), a TAG permite um mapeamento das intenções do usuário (tarefa) em procedimentos do sistema, além da especificação desde a intenção do usuário até o nível de ativação do sistema, registrando adequadamente a tradução semântica entre a intenção do usuário e o procedimento do sistema. Porém, a TAG não registra adequadamente a expressão

do modelo funcional do sistema, porque não registra o retorno do sistema e conseqüentemente o comportamento das primitivas, além de não registrar a sobreposição de recursos expressivos, como uma alteração de cor e forma sobre um objeto (Martins & De Souza, 1998). Em razão disto, com somente o uso da TAG perdem-se ocorrências que caracterizam diferenças de comportamento entre primitivas.

Ainda segundo Martins em (Martins & De Souza, 1998), parte dos problemas da TAG decorrem especificamente de sua aplicação à LV, outra parte é intrínseca à própria estrutura da gramática. A proposta de Martins em (Martins & De Souza, 1998) é que a notação STAG complemente a TAG, superando as deficiências recém citadas.

A notação da STAG incorpora implicitamente alguns princípios básicos da ES, de forma que estes princípios são automaticamente respeitados, pela simples utilização da mesma (Martins & De Souza, 1998). A STAG tem como objetivo atender às necessidades de projeto e análise. Quando utilizada para projeto, procura auxiliar o projetista na definição da linguagem de interface dentro da proposta da ES, observando requisitos específicos para LVs, em particular a consistência (Martins & De Souza, 1998). A consistência é difícil de ser definida ou medida, mas é reconhecida como sendo a maior determinante de legibilidade (Payne & Green, 1986). A vantagem da consistência está na facilidade de generalizações por parte do usuário, que tendo aprendido algumas partes do sistema pode inferir outras.

Quatro partes compõem a STAG (Martins & De Souza, 1998): (a) a declaração de tipos e instâncias, (b) o mapeamento entre tipos e instâncias de conteúdo e expressão, (c) um dicionário de tarefas simples, opcional, e (d) um conjunto de regras chamado de "gramática".

(a) A declaração de tipos e instâncias é feita de acordo com a segmentação dos contínuos do conteúdo e da expressão, de usuário e sistema. Onde para cada tipo é feita a declaração de instâncias. Os critérios utilizados para a segmentação são os citados em Martins (Martins, 1998) capítulo 4 e contém as etapas: (a) definição de tipos de conteúdo do sistema, onde são declaradas as entidades correspondentes às primitivas semânticas do sistema, seus atributos e relacionamentos; (b) definição de tipos da expressão do sistema, onde os tipos são estabelecidos em função da análise do dispositivo de comunicação, no caso o vídeo; (c) definição de tipos de conteúdo do usuário, que são os tipos de ação que o usuário pode expressar considerando interfaces por menus e por MD; definição de tipos da expressão do usuário, no caso os sinais emitidos pelo mouse.

(b) O mapeamento entre tipos e instâncias de conteúdo e expressão faz a correlação entre os planos de conteúdo e expressão. Esta correlação geralmente é realizada por um conjunto de regras que geram os tipos gerais da linguagem. O sinal " $\leftrightarrow$ " indica o mapeamento entre o conteúdo e sua expressão.

(c) O dicionário de tarefas simples lista as tarefas qualificadas por elementos do modelo adotado para a segmentação, onde cada elemento do modelo que particulariza a tarefa é denotado por um termo descritivo. O sinal "=" associa o termo descritivo a tipos/instâncias do conteúdo. O conjunto de informações vem dentro de colchetes, e as informações vêm separadas por vírgulas.

(d) As regras gramaticais são compostas por: (a) lado esquerdo, que é um elemento simples consistindo de um rótulo arbitrário e um conjunto



de características contidos em colchetes, os quais podem conter qualquer número de informações que particularizam a tarefa, valoradas ou não valoradas, (b) o sinal de “→”, (c) lado direito, que contém uma seqüência ordenada de termos, descritos da mesma forma que o lado esquerdo, e símbolos terminais. Esta gramática segue os termos de gramática gerativa e sua saída é uma lista de tipos e instâncias do conteúdo utilizados por usuários e sistemas durante a execução da tarefa.

As evoluções da STAG a partir da TAG referem-se à dificuldade da TAG em caracterizar os elementos gramaticais, uma vez que não distingue os planos de expressão e do conteúdo, conforme o enfoque da Semiótica, e, por esta razão, não permite a associação destes planos nos níveis de associação por tipo e por instância. Isto porque em linguagens textuais não existe a necessidade de escolha entre tipos expressivos em função de tipos de conteúdo (Martins & De Souza, 1998).

O presente trabalho tem por objetivo a proposta de uma interface visual adequada para consulta a base de dados, logo é de suma importância a geração de uma linguagem visual que seja sistêmica (Martins & De Souza, 1998), além de uma gramática subjacente para orientar o usuário na sistematização das tarefas a serem executadas durante a interação com o sistema. Com o uso da ferramenta STAG é possível aliar a descrição da LV e da gramática necessária para a proposta da interface, objeto deste estudo.

#### 3.1.4 Considerações sobre os Modelos de Tarefas

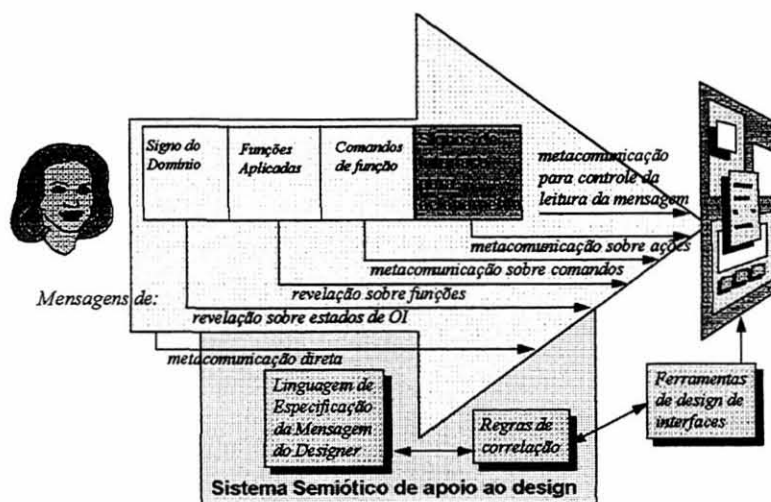
Segundo Leite em (Leite, 1998), os modelos de tarefas recém apresentados muitas vezes revelam quais as estruturas mais adequadas para o modelo de interação, tais como vocabulário significativo, interação baseada no *display*,

consistência, entre outros. Porém, apesar de eles se relacionarem diretamente com o modelo de interação e poderem ser aplicados no processo de design, ainda são soluções parciais para o problema do mapeamento tarefa-ação. Logo, a abordagem cognitiva não deve ser a única considerada no processo de desenvolvimento da interface e do sistema, uma vez que signos e processos de signos, que são o princípio subjacente à interação humana com sistemas computacionais, também devem ser considerados.

Isto posto, o designer precisa, segundo Leite e de Souza em (Leite & De Souza, 1999), complementar esta especificação considerando que estes componentes são parte do processo comunicativo existente entre o designer e o usuário. Ele deve realizar a especificação de cada componente do modelo de usabilidade como parte da mensagem global que é enviada ao usuário. Esta especificação deve ser abstrata uma vez que a mensagem é descrita independentemente de como será a sua expressão final, ou seja, de quais serão os *widgets* utilizados para expressá-la. Usando a LEMD, descrita na seção 2.5.1, é possível descrever a mensagem de maneira abstrata e estruturada. A especificação da mensagem do designer inclui a descrição das funções da aplicação, dos signos do domínio e dos comandos sem a preocupação com quais *widgets* e configurações espaciais e temporais elas serão expressas.

Por fim, a expressão concreta da mensagem deve ser concebida e especificada para que possa ser apresentada ao usuário durante o processo de interação. Na figura 6, pode ser visualizado o processo de design de interfaces compreendendo as etapas recém descritas e os principais formalismos e ferramentas de apoio a este processo.

Figura 6: Formalismos de apoio ao design



Fonte: Leite, 1998

## **4. FERRAMENTAS DE CONSULTA A BASE DE DADOS**

Uma dificuldade tradicional no acesso a bancos de dados tem sido causada pela necessidade de utilização, pelo usuário final, de linguagens de consulta textuais, tais como SQL, e de prévio conhecimento do esquema de base de dados. Este esforço tenta ser minimizado por interfaces visuais para banco de dados que objetivam aproximar os esquemas que descrevem os dados dos sistemas de informação dos conceitos do mundo real, percebidos pelo usuário. Neste sentido este capítulo procura situar o leitor nos conceitos que dão o suporte necessário à representação visual e às tarefas solicitadas pelo usuário que é dado pelo modelo de dados, que, por sua vez, deve ser capaz de capturar os conceitos da base de dados sobre o qual a interface atua. Também é apresentada a facilidade proporcionada pela abordagem de banco de dados visuais, que envolve objetos gráficos tais como diagramas e ícones, acrescidos das ações do usuário sobre esses, isto é, de operações sobre as representações para interagir com a base de dados. Como um grande número de ambientes de consulta para base de dados têm sido proposto com esta finalidade, este capítulo aproveita alguns destes ambientes para realizar uma análise em seus pontos positivos e negativos à luz dos conceitos de IHC apresentados nos capítulos anteriores.

### **4.1 O MODELO DE DADOS**

Independentemente da interface-usuário utilizada, existe um sistema gerenciador de banco de dados que irá fazer a tradução das intenções expressas pelo usuário na interface e o sistema computacional que opera os dados

armazenados em uma base de dados. Uma base de dados pode ser descrita em termos de um modelo de dados, que seria um conjunto de conceitos para organizar um domínio de interesse, de forma que possam ser processados por um SGBD. Pode-se categorizar estes modelos de acordo com os conceitos que eles utilizam para descrever a estrutura de um banco de dados (Angelaccio et al., 1990 (b)).

Em particular, o modelo Entidade Relacionamento (ER), em suas várias formas estendidas, é o mais utilizado. Isto é consistente com o atual estado dos negócios, nos quais o modelo ER age como um padrão na área de métodos de projetos conceituais e ferramentas de bancos de dados. Os modelos de dados existentes fazem uso de conceitos similares para representar a informação, dos quais podem ser citados: (Catarci et al., 1997).

- objetos do mundo real, agrupados a suas propriedades particulares. Exemplo: A idade de Charles é 40 anos.
- Relacionamentos entre estes objetos. Exemplo: Charles é casado com Ana.
- Classes de objetos. Exemplo: a classe de pessoas, classe de animais, classe de cidades.
- Relacionamentos entre classes de objetos. Exemplo: pessoas vivem em cidades.

Os objetos e seus relacionamentos constituem o que é chamado de extensão da base de dados, ou somente extensional, enquanto que as classes de objetos e seus relacionamentos formam as intenções da base de dados, também chamadas de esquema da base de dados ou simplesmente intenção da base de dados (Catarci et al., 1997).

A maioria das metodologias atuais utiliza uma modelagem de dados semântica, logo o conhecimento da semântica de dados é necessário. Quando usamos um modelo de dados semântico, é possível que diferentes projetistas modelem o mesmo objeto do mundo real de diferentes formas. Isto pode acontecer porque os mesmos dados podem ser modelados de maneiras equivalentes ou porque os projetistas têm percepções diferentes daquela realidade. Este fato, a multiplicidade de possíveis representações de um determinado objeto no mundo real, é chamado de “semântica relativista” (Spaccapietra & Parent, 1992).

#### 4.1.1 O Modelo ERC+

O modelo escolhido neste trabalho é o modelo ERC+ (Entidade Relacionamento Complexo Estendido). Este modelo é um modelo ER estendido que foi definido para apoiar a descrição de objeto complexo e sua manipulação. É chamado ERC+ devido ao seu significado: ER para objetos Complexos (o “+” denota o enriquecimento do modelo ERC básico) (Spaccapietra & Parent, 1992). Pelo fato de o modelo ERC+ ter sido especialmente desenvolvido para suportar objetos complexos e suas identidades, além de suportar a semântica relativista e, por esta razão, expressar, com maior riqueza, a relação existente entre o significado do objeto no mundo real e a sua representação no banco de dados, ele está em conformidade com as necessidades de uma ferramenta mais próxima do modelo do usuário, o que auxilia no desenvolvimento de uma interface-usuário para a geração de consulta visual em um banco de dados.

A modelagem do objeto complexo e seu gerenciamento são as metas principais dos modelos de dados de hoje. Através de um objeto complexo, quer se

transmitir a idéia de que um objeto é representado por uma coleção de informações, seus componentes, tal que cada um destes componentes pode ser representado por uma coleção de informações, e assim por diante. O suporte a objetos complexos não contradiz a distinção básica que o modelo ER faz entre entidades e atributos: esta distinção está baseada em considerações semânticas (que são os objetos primários de interesse), não em propriedades sintáticas (sendo atômico ou não) (Spaccapietra & Parent, 1992).

O modelo ERC+ permite especificamente esta descrição interativa de um objeto, até um número arbitrário de níveis. A estrutura resultante é uma árvore de atributo cuja raiz é o objeto. Além disso, qualquer nó na árvore pode levar a um valor de atributo sem igual, ou um multi-conjunto de valores de atributo. Além disto, o modelo ERC+ usa o conceito de tipo entidade para descrever objetos e o de tipo relacionamento para representar relacionamentos entre objetos. Sendo que a estrutura de uma entidade consiste em um conjunto de um ou mais atributos. E os relacionamentos podem unir qualquer número de entidade. É dito que eles são cíclicos se a mesma entidade participa, mais de uma vez, no relacionamento. Então um papel é associado a cada participação de uma entidade em um relacionamento. Este papel é caracterizado por seu mínimo e máximo de cardinalidades, especificadas como 0-1, 0-n, 1-1 ou 1-n de acordo com o vínculo da entidade para o relacionamento. (Gueiber, 2001)

Os atributos podem ser (Spaccapietra & Parent, 1992):

- Obrigatórios ou opcionais: uma instância de um atributo opcional pode estar vazia (nenhum valor), para um atributo obrigatório um valor deve ser definido, em cada instância do atributo.

- Monovalorado ou multivalorado: uma instância de um atributo multivalorado pode incluir vários valores, enquanto uma instância de um atributo monovalorado é composta de um único valor.
- Simples ou complexos: se simples, o atributo possui domínio definido e indica que o mesmo é o último componente da hierarquia de um grafo, ou seja, uma folha. Se complexo, o atributo é composto de um conjunto de outros atributos que são os componentes daquele atributo. Os atributos componentes podem ser simples ou complexos. Agrupando-se esta definição, pode-se proceder a qualquer número de níveis.

As entidades e relacionamentos podem ter zero, um ou mais conjuntos de atributos que servem como identificadores. Já um atributo identificador (*object identifier* - oid) indica que o valor deste atributo é distinto para cada instância desta entidade, sendo que o mesmo é utilizado como identificação da instância (Spaccapietra & Parent, 1992).

Algumas vezes é necessário formar um identificador composto, o que quer dizer que o identificador é formado por mais de um atributo, sendo que a combinação destes atributos deve ser única na entidade. Se nenhum identificador é conhecido, a população respectiva pode incluir duplicidades (ocorrências diferentes com o mesmo valor) (Spaccapietra & Parent, 1992).

Duas generalizações são suportadas no modelo ERC+, "*is-a*" e "*may-be-a*". A generalização "*is-a*" corresponde ao conceito de generalização onde se identifica uma entidade genérica que possui as características principais de entidades específicas. Por exemplo, na figura 7 podemos definir uma entidade genérica

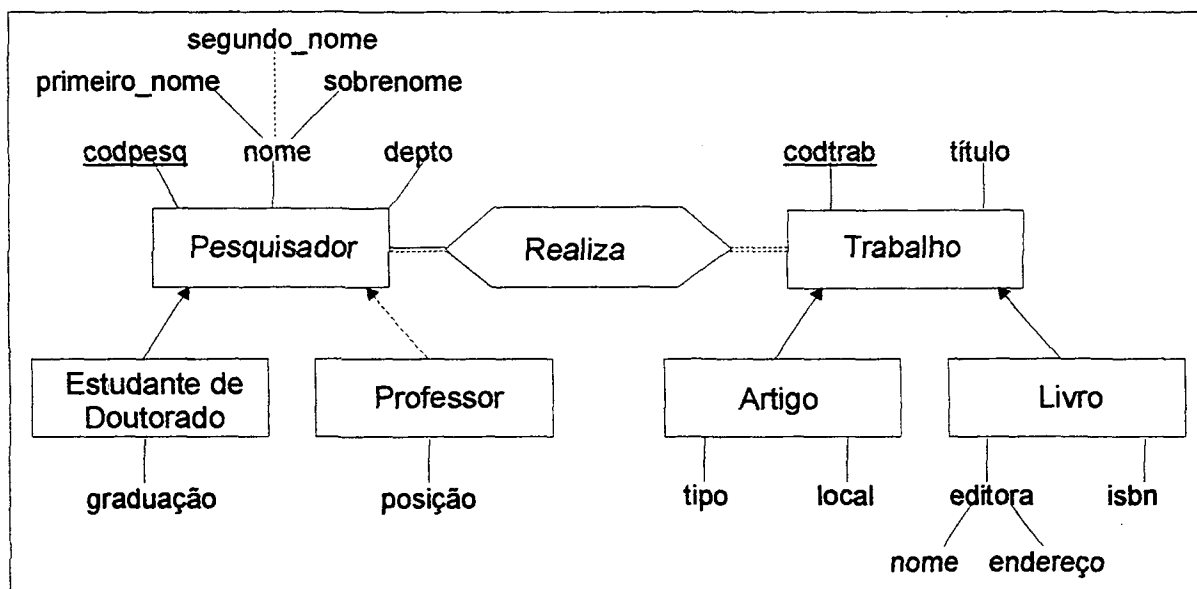


Estudante que indica uma generalização das entidades específicas Estudante\_graduação e Estudante\_pósgraduação.

A generalização “*may-be-a*” tem semântica semelhante, mas não requer uma dependência de inclusão entre o subtipo e o tipo. Este conceito de generalização “*may-be-a*” indica que a população do tipo entidade genérico não inclui o conjunto da população do tipo entidade específico, ou seja, uma consulta realizada no tipo entidade específico não abrange os dados do tipo entidade genérico (Spaccapietra & Parent, 1992).

Quando existe a ocorrência de uma generalização, os atributos da entidade pai são comuns às entidades filhas, sendo que os atributos específicos de cada entidade são representados nas entidades filhas (Spaccapietra & Parent, 1992).

Figura 7: Objetos representados de acordo com o modelo ERC+



Fonte: (Rodacki, 2000)

A partir da figura 7, que mostra um esquema ERC+, pode-se abstrair a representação utilizada (Rodacki, 2000):

- Uma entidade é representada por um retângulo, sendo que o nome da entidade é inserido dentro do retângulo, com a inicial do nome em letra maiúscula.
- Um atributo é representado em letras minúsculas com uma linha simples contínua unindo a sua entidade pai, se o atributo for obrigatório e uma linha simples tracejada, se o atributo for opcional. Os atributos sublinhados indicam atributos identificadores.
- Um relacionamento é representado por um losango com o nome do relacionamento em seu interior, com a inicial em letra maiúscula. Uma linha simples contínua identifica um relacionamento monovalorado e mandatário, 1:1; uma linha simples tracejada indica um relacionamento opcional monovalorado, 0:1; uma linha dupla tracejada representa um relacionamento multivalorado opcional, 0:n; e um relacionamento representado por uma linha tracejada e uma linha contínua indica uma representação mandatária multivalorada, 1:n.
- Uma generalização “is-a” é indicada através de uma linha contínua com uma flecha apontando para a entidade genérica. Sendo que uma linha tracejada apontando para a entidade genérica indica uma generalização “*may-be-a*”.
- Como pode ser visto na figura 7, a entidade Pesquisador está ligada à entidade Trabalho pelo relacionamento Realiza, que possui a seguinte semântica: “um pesquisador pode realizar zero ou mais trabalhos”, e, “ um trabalho pode ser realizado por um ou mais pesquisadores”.
- A entidade Pesquisador é uma generalização das entidades Estudante de Doutorado e Professor, onde a flecha formada por uma linha contínua indica

uma generalização “*is-a*”, que dita que todos os estudantes de doutorado são pesquisadores. A generalização indicada por uma flecha formada por uma linha tracejada indica uma generalização “*may-be-a*” que diz que os professores podem ou não ser pesquisadores. Por exemplo, no caso de um professor adjunto, onde o mesmo pode não exercer uma atividade de pesquisa.

- Ainda na entidade Pesquisador, podemos verificar a existência de um atributo complexo, nome, que possui atributos obrigatórios, primeiro\_nome e sobrenome, e um atributo opcional segundo\_nome.
- A entidade Trabalho é uma generalização das entidades Artigo e Livro, onde todos os artigos e livros são considerados trabalhos (generalizações “*is-a*”).
- Os atributos codpesq e codtrab são mostrados sublinhados por serem atributos chaves das entidades Pesquisador e Trabalho, respectivamente.

O modelo ERC+ é complementado com definições formais através de uma linguagem de manipulação, com uma álgebra associada para que seja possível manipular consultas em um banco de dados ERC+ (Spaccapietra & Parent, 1992).

#### 4.1.2 A Álgebra do Modelo ERC+

Um dos principais problemas das linguagens de manipulação de dados desenvolvidas para trabalhar com modelos orientados a objetos e extensões do modelo entidade relacionamento é não dispor de operadores para trabalhar com todos estes conceitos, especialmente operadores para manipular diretamente

relacionamentos genéricos. A álgebra ERC+ foi construída para manipular todos os conceitos do respectivo modelo. (Spaccapietra & Parent, 1992).

Nesta seção serão descritos os operadores de consulta associados com o modelo ERC+, conforme especificado em (Spaccapietra & Parent, 1992) e utilizado em (Dennebouy et al., 1995) e (Gueiber, 2001).

A álgebra inclui operadores primitivos:

- *Selection*: que é a seleção de objetos baseados em um dado predicado;
- *Projection*: que é a projeção de objetos em um sub-conjunto de seus atributos;
- *Union*: que é a união de populações compatíveis, aproximadamente, em um mesmo esquema;
- *Relationship-join (r-join)*: é o operador que agrupa dentro de uma entidade simples a informação dispersa sobre as entidades ligadas através de um relacionamento;
- *Identity-join*: o operador *i-join* permite juntar objetos participantes em uma relação multi-instanciada;
- *Product*: é o operador de produto, similar ao *r-join*, mas que associa a cada ocorrência do primeiro objeto todas as ocorrências do segundo objeto;
- *Compression*: é um operador específico do ERC+, que permite descartar duplicidades em uma população de um objeto;
- *Reduction*: também é um operador específico do ERC+, que permite a eliminação de valores de um atributo que não satisfaz a um determinado predicado.

A álgebra ERC+ também contém dois operadores sintáticos, *renaming* e *simplification*, o qual permite adequar o esquema de uma entidade com as regras do modelo ou da álgebra:

- operador *renaming* muda o nome de um atributo para preparar a fusão de atributos similares durante uma união;
- operador *simplification* exclui complexidade desnecessária na estrutura que pode ser construída por outros operadores, a projeção em particular. A simplificação exclui um nível na estrutura complexa quando um atributo complexo tem somente um atributo componente (exceto se são ambos multivalorados).

Operadores derivados podem ser definidos para auxiliar o usuário a escrever consultas mais curtas.

- operador *difference* (-) pode ser definido através da composição apropriada do *product*, *selection* e *projection*.
- operador *intersection* pode também ser definido através de duas diferenças.

Comumente a semântica destes operadores é formar uma nova população correspondente aquelas ocorrências dos primeiros operandos para os quais não há ocorrências no segundo operando com mesmo oid - diferença; ou para aquelas ocorrências dos primeiros operandos para os quais há uma ocorrência no segundo operando com o mesmo oid - interseção.

No modelo ERC+ ainda são possíveis dois tipos de ligações que precisam ser diferenciados *is-a* e *may-be*:

- ligação *is-a* - dada uma ligação entre duas relações  $R_1$  e  $R_2$ , que possuem a mesma composição de chave primária, cujos valores estão contidos nos conjuntos  $PK_1$  e  $PK_2$ , se  $PK_2 \subseteq PK_1$  a ligação é traduzida para o tipo *is a*. Cabe observar que a declaração formal no dicionário de dados de uma chave estrangeira, cuja composição é a mesma de sua chave primária leva a esta mesma conclusão;
- ligação *may-be-a* - considerando uma associação entre duas relações  $R_1$  e  $R_2$ , que possuem a mesma composição de chave primária, cujos valores estão contidos nos conjuntos  $PK_1$  e  $PK_2$ , se  $PK_2 \not\subseteq PK_1$  entretanto  $PK_1 \cap PK_2 \neq \emptyset$ , o relacionamento é traduzido para *may be a*. Este tipo de ligação não permite a inclusão de restrição como chave estrangeira no dicionário de dados e normalmente é validada pela aplicação, assim para determinar esta forma de ligação deve ser feita análise das instâncias.

Cada operação citada resulta na criação de novos objetos, com seus atributos, ambientes e população derivados das operações através de suas regras específicas. Estas operações também podem ser combinadas em expressões complexas.

ERC+ incorpora duas linguagens básicas: a álgebra apresentada acima, e um cálculo. Estas linguagens básicas são as fundamentações básicas sobre as quais linguagens amigáveis ao usuário tais como SQL e linguagens gráficas são construídas (Spaccapietra & Parent, 1992).

## 4.2 FORMULAÇÃO DE UMA CONSULTA EM UMA BASE DE DADOS

Existem duas formas de se extrair dados de uma base de dados. Uma delas é extensional, onde se navega através da base de dados até o nível de ocorrências, selecionando os dados de interesse. O termo “navegar” está relacionado a aspectos relativos a consulta, como um típico operador da consulta utilizado para: selecionar elementos visuais, capturar elementos adjacentes e a criar ligações entre elementos não conectados. Os *browsers* são as ferramentas utilizadas para realizar esta navegação (Dennenbouy, 1995).

A outra forma é intensional, onde se formula uma consulta declarando quais propriedades devem ser alcançadas pelos objetos e/ou valores em uma base de dados para que sejam selecionados e incluídos no esquema da base de dados que irá definir quais classes de objetos são relevantes em termos de consulta e em termos de valores de atributos e relacionamentos entre objetos, para selecionar o sub-conjunto de população relevante, gerando o sub-esquema da consulta. Uma consulta também define quais dados de um sub-conjunto relevante devem ser colocados na resposta (uma operação de projeção) e como estes dados devem ser estruturados no resultado final (a não ser que o resultado seja, por definição, um relacionamento de primeira ordem). As etapas descritas podem ser visualizadas na figura 8 (Angelaccio et al., 1990 (b)).

Em uma linguagem clássica de consulta como SQL, estas duas formas de consulta são expressas em blocos *SELECT-FROM-WHERE*: a cláusula *FROM* define as classes relevantes; a cláusula *WHERE* restringe estas classes aos sub-conjuntos relevantes; e a *SELECT* especifica a projeção do dado (Dennenbouy, 1995).

Em um ambiente visual, a definição de uma classe de objetos relevantes (*FROM*) é realizada pela visualização do esquema da base de dados na tela (em forma de diagrama ou de lista de menu) e pressionando o botão esquerdo do mouse no tipo desejado para inseri-lo no sub-esquema de consulta. O sub-esquema da consulta é um sub-grafo do grafo original do esquema. Esta é a etapa inicial do processo da elaboração de uma consulta em toda linguagem de consulta visual. Nesta etapa é feita a redução do esquema para que ele contenha somente os elementos que estarão envolvidos na consulta (Dennenbouy, 1995).

A definição dos sub-conjuntos relevantes das classes na consulta (*WHERE*) é expresso em forma de condições (predicados) em instâncias dos tipos no sub-esquema. Somente os objetos que atendem às condições irão fazer parte do resultado. Nesta fase a interface formula a consulta em termos do objetivo da SGBD, que normalmente é expressa em SQL.

A definição de quais dados serão apresentados ao usuário, e da forma como serão apresentados (*SELECT*) gera uma nova definição, um objeto virtual, que é o resultado da consulta realizada (Dennenbouy, 1995).

Dentro das etapas do processo de geração de uma consulta este trabalho irá se dedicar à primeira destas fases onde serão definidas quais entidades são relevantes em termos de consulta e em termos de valores de atributos e relacionamentos entre entidades, para selecionar o sub-conjunto de população relevante, isto é, a seleção do sub-esquema da consulta (figura 8). As outras etapas serão objeto de outros trabalhos relacionados.



#### 4.3 A INTERFACE NO PROCESSO DE CONSULTA EM UMA BASE DE DADOS

Na maioria das ferramentas disponíveis de linguagem visual para tratamento de base de dados, o processo de construção da consulta, dividido nas fases recém citadas, faz uso de objetos gráficos que são manuseados pelo usuário através do mouse, o que caracteriza a manipulação direta (Hutchins et al., 1986) de objetos. No domínio de banco de dados os objetos que são manipulados significam uma estrutura de dados chamada e armazenada como uma unidade de base de dados, a qual é recuperada e manipulada pelo usuário.

Desde que interfaces WIMP (*window, icon, menu e pointers devices*), interfaces que integram recursos de manipulação direta e interfaces por menus, foram aproveitadas para consultas a banco de dados, os operadores da linguagem de consulta precisaram ser adaptados. Tal modelo de interface não poderia ser aplicado sobre o conceito formal de banco de dados, mas sim sobre uma representação externa apropriada de tais conceitos, de forma a permitir que os mesmos se tornassem fáceis de serem percebidos e interpretados pelo usuário. Desta forma a representação deveria ser parecida com a realidade do usuário. Quando a representação escolhida para a realidade é a visual pode-se dizer que está sendo utilizada uma linguagem visual de consulta (*Visual Query Language – VQL*) (Catarci et al., 1997).

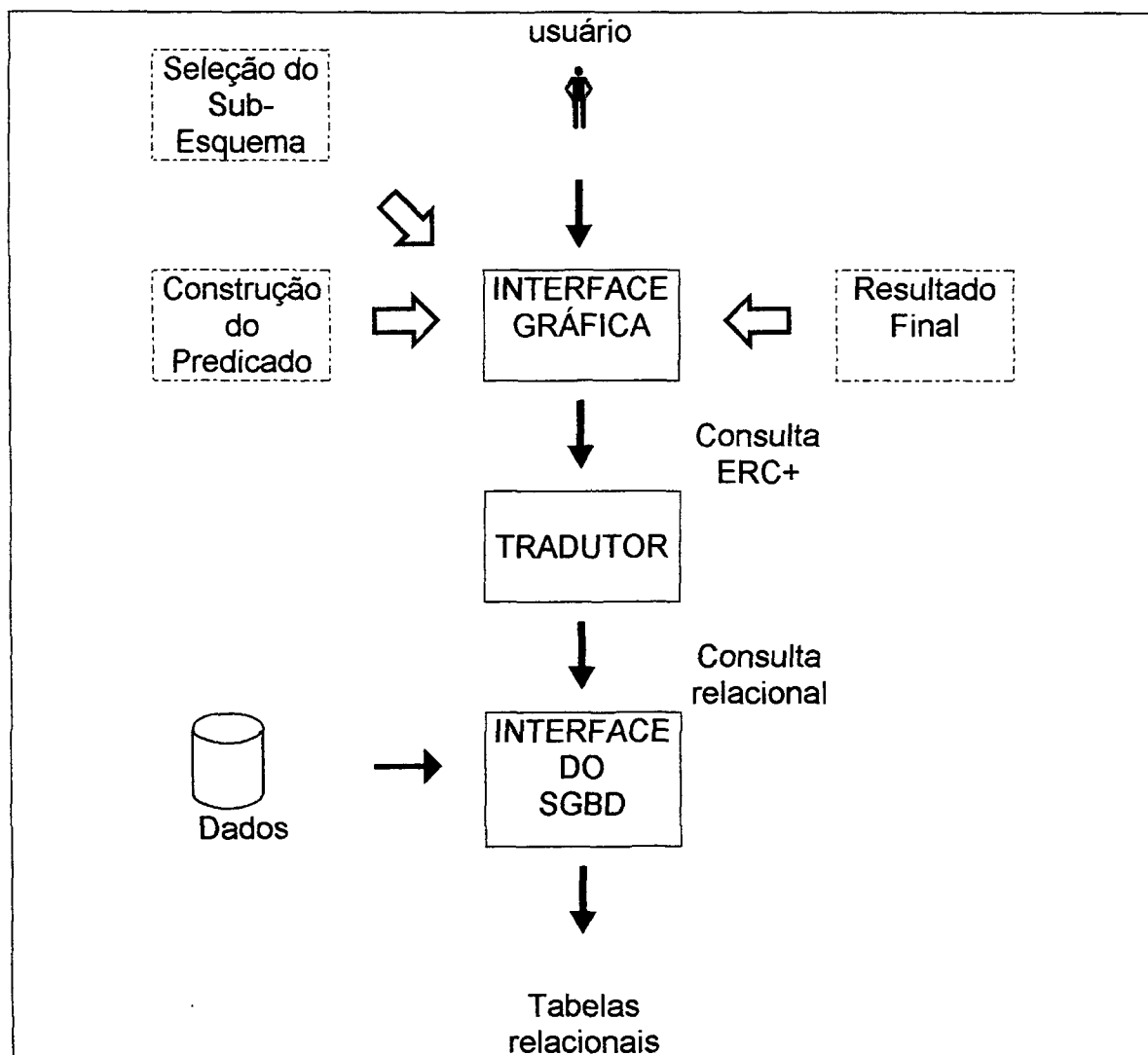
As VQLs são linguagens implementadas para serem utilizadas em sistemas de consulta visual (*Visual Query Systems – VQS*). Tais linguagens compõem uma sub-classe particular da classe mais geral, de linguagens visuais (*Visual Languages – VL*), e têm o objetivo de facilitar a extração de informações relevantes de uma base de dados. Deve ficar claro que as linguagens visuais de consulta são linguagens cuja construção é visual, mas que podem manipular também objetos que

não necessariamente têm uma representação visual, como, por exemplo, representações alfanuméricas em um banco de dados tradicional (Catarci et al., 1997).

O objetivo principal de se adotar uma representação visual em um sistema de consulta é o de comunicar de forma clara ao usuário as informações contidas na base de dados, concentrando-se em características essenciais e omitindo os detalhes que não têm interesse para o usuário. Esta informação é estruturada internamente de várias maneiras dependendo das características do modelo de dados utilizado, mas deve ser traduzida para o usuário de forma clara, de maneira a facilitar a compreensão. Entretanto, a representação visual da interface deve ser mapeada em termos dos conceitos internos da base de dados, uma vez que tais comandos devem ser manipulados pelo Sistema Gerenciador de Banco de Dados. Para combinar os dois lados, do usuário e do sistema, a representação deve ser realizada por meio de um formalismo visual. Formal, porque os dados devem ser manipulados, mantidos e analisados por programas que executam em computadores e visual, uma vez que deverão ser compreendidos e manipulados por humanos, os usuários. O formalismo visual inclui objetos familiares ao usuário, tais como: tabelas, diagramas, ícones, entre outros. Eles são componentes da representação visual proposta pela VQS (Catarci et al., 1997).

Porém todo o processo de gerenciamento da base de dados é realizado por um SGBD que utiliza um modelo de dados específico. Logo a VQL é uma linguagem visual que está presente na interface do sistema que é apresentada ao usuário, motivada pela necessidade de redução do esforço cognitivo do usuário para entender e utilizar a linguagem do domínio do SGBD, como apresentado na figura 8:

Figura 8: Arquitetura de um ambiente de VQS baseado em Angelaccio et al., 1990 (b)



#### 4.4 PANORAMA GERAL DE FERRAMENTAS DE CONSULTAS A BASE DE DADOS

O incentivo às pesquisas na área de Banco de Dados ocorreu com o desenvolvimento da abordagem de Orientação a Objetos (OO), que permite uma apresentação alternativa para o formato planar de manuseio relacional das estruturas de dados. Desta forma, objetos complexos puderam ser definidos e manipulados. Por outro lado, a abordagem de banco de dados invadiu novas áreas de aplicações, com muitos usuários potenciais que normalmente resistem em

visualizar suas informações em forma de uma estrutura planar, mesmo que a tecnologia os obrigue a isto. O suporte a objetos complexos levou à necessidade de novas interfaces e metodologias de projetos de formulação de consulta em banco de dados (Dennenbouy, 1995).

Baseados nesta evolução e considerando, por sua vez, a complexidade das linguagens textuais de consulta, que excedem a capacidade da maioria dos usuários, vem surgindo um número crescente de interfaces visuais que suportam definição e manipulação de objetos complexos. Como a idéia é ter-se um sistema que seja de “fácil entendimento” para o usuário leigo no domínio de Ciência da Computação, a linguagem textual apresentada pela ferramenta SQL resulta insatisfatória.

A partir de estudos realizados recentemente, derivados da necessidade de usuários leigos terem necessidade de operar ferramentas de SGBD em suas atividades profissionais, chegou-se a uma ferramenta que procura aproximar, um pouco mais, o usuário do sistema, que é a QBE (*Query By Example*) (Zloof, 1977). Mas a QBE ainda está longe de ser o que se entende por uma ferramenta “amigável”. Assim, procurou-se aliar a tecnologia de desenvolvimento do hardware, onde os computadores começaram a poder expressar-se em duas dimensões, ao uso de estruturas gráficas para representar objetos de seu interesse. Esta evolução permitiu a criação de ferramentas que fazem uso do recurso de manipulação direta (Hutchins et al., 1986) de gráficos, onde o usuário movimentava com o mouse os objetos representados graficamente existentes na tela do computador.

Dentre as ferramentas de consultas visuais mais avançadas estão Pasta-3, QBD\* e Super. A manipulação visual de dados nestas ferramentas está atrelada a algum gerador SQL, que opera em uma base de dados relacional e permite a

construção de consultas utilizando menus para selecionar as tabelas e as operações que serão realizadas sobre estas tabelas. Em razão disto elas têm algumas falhas no que diz respeito ao suporte aos paradigmas dos objetos uma vez que a estrutura relacional da base de dados escolhida influencia a linguagem. Elas também não têm uma abordagem totalmente visual: elas têm um formalismo visual para os operadores principais, em uma metodologia procedural, e não em uma especificação visual, que pode ser chamada “declarativa”, por não utilizar um mapeamento um-para-um na álgebra escolhida. Isso se soma ao fato de que elas têm dado pouca atenção ao complexo processo de formulação de consultas, resultando em poucas facilidades para correção das consultas realizadas (Dennenbouy, 1995). Isto leva a concluir que o projeto de interfaces visuais para elaboração de consultas a banco de dados ainda é uma área de interesse de pesquisa. Enquanto algumas ferramentas oferecem facilidades de consulta visual e linguagens textuais de formulação de consultas, quase nenhuma dá suporte à completa formulação visual de consultas.

#### 4.5 ANÁLISE DAS FERRAMENTAS PASTA3, QBD\* E SUPER

##### 4.5.1 Pasta-3

Conforme descrito por Kuntz e Melchert em (Kuntz & Melchert, 1989), a ferramenta Pasta-3 apresenta características que contribuem para os novos conceitos de manipulação direta utilizando recursos gráficos (GrDM – *Graphic Direct Manipulation*), contemplando uma base gráfica de manipulação direta com representação icônica da expressão da consulta, um ambiente cooperativo permitindo a complementação automática de caminhos e reedição de consultas, além de seu poder de expressão que inclui o envolvimento de códigos em Prolog e recursividade. Apesar disto a abordagem básica do Pasta-3, não tem intenção de

definir uma nova linguagem formal, mas prover uma sofisticada capacidade de manipulação direta através de recursos gráficos para uma linguagem formal já existente. O Pasta-3 inclui capacidades de GrDM para todas as atividades pertencentes a um DBMS (*Data Base Mangement System*).

O Pasta-3 é uma interface para o KB2, um sistema de base de conhecimento para Prolog, isto é, um sistema de base de dados lógico que suporta tanto *loose coupling* como a integração do Prolog com um modelo relacional DBMS. O KB2 utiliza o modelo Entidade-Relacionamento, estendido com herança e regras dedutivas. Ele foi desenvolvido por Mark Wallace e outros membros do *Knowledge Base Group*.

O Pasta-3 trabalha com diferentes tipos de janela na tela, apresenta o esquema selecionado utilizando o diagrama ER que é mostrado através de um grafo de entidades e seus atributos, onde o relacionamento é demonstrado através de linhas que conectam as entidades. Para a seleção de sub-esquema as entidades e relacionamentos podem ser selecionadas por meio de um menu de consulta que aparece no topo da janela de consulta, ou valendo-se de *copy/paste* de trechos do esquema original. O usuário pode, por meio do menu do topo da janela de consulta, selecionar os atributos. Somente estarão disponíveis para seleção os atributos pertencentes às entidades e relacionamentos selecionados previamente. Este tipo de recurso contribui para a diminuição de erros, onde o usuário somente pode selecionar os itens disponíveis, remetendo ao processo de WYSIWYG.

Na medida em que o usuário seleciona os elementos de interesse para sua consulta é montado, em uma janela separada, um "esqueleto" da consulta no formato de tabela, onde o usuário acompanha o que está ocorrendo e valida se sua meta está sendo atingida. Assim, o usuário consegue visualizar a formação de sua

consulta de uma forma comum para ele, que é a leitura de tabelas, utilizando uma metáfora do mundo real. Nesta janela, o usuário consegue diferenciar entidades e relacionamentos de atributos de uma forma visual, pois as entidades e relacionamentos aparecem em um retângulo com bordas duplas, enquanto que os atributos aparecem em retângulos com bordas simples.

Nesta tabela existe uma linha em branco embaixo de cada atributo onde pode ser descrito o conteúdo desejado para realizar a consulta. Para facilitar ao usuário diminuindo a necessidade de memorização de valores dos atributos, são apresentados, por meio de um item de menu, os valores possíveis de um atributo, quando este não ultrapassar a quantidade de 15 itens. Desta forma, é possível diminuir o número de respostas vazias às consultas quando um valor desconhecido é colocado pelo usuário. O Pasta-3 procura encorajar o uso deste recurso, mas não impõem este recurso ao usuário. Também é possível utilizar o recurso de *copy in value*, que pode ser usado quando o valor já está na tela, mas em outro lugar, como resultado de uma consulta anterior ou de um *browsing*. Desta forma, a possibilidade de erros de digitação é excluída, porém usando este recurso não existe como assegurar a existência de respostas vazias, uma vez que não existem regras para restringir de onde o usuário pode realizar a cópia.

Outro fato interessante ocorre quando o usuário seleciona diversas entidades e relacionamentos, mas esquece de selecionar um determinado relacionamento crucial para estabelecer o sub-esquema desejado. Porém o sub-esquema deve ser considerado correto, uma vez que, tradicionalmente, os usuários têm problemas com a semântica, no que tange a selecionar a conexão correta, e problemas sintáticos para construir as expressões utilizando texto linear. Nestes casos, o Pasta-3 irá encontrar os itens faltantes e oferecer ao usuário para que os adicione, caso tenha

faltado somente um caminho para conectar, ou propor uma escolha, caso exista possibilidade de ambigüidade na seleção do caminho.

O uso de recursividade nesta ferramenta é solucionado por meio da repetição do elemento recursivo, fazendo uso dos conceitos de elementos predecessor e sucessor, para que a recursão seja realizada de forma correta. Isto ocorre por meio de diálogo entre o sistema e o usuário. Considerando que para um usuário leigo o conceito de recursividade não é natural, o recurso de repetição do elemento facilita o entendimento, principalmente, pelo fato de o usuário ser guiado para tal, sem perceber, valendo-se do recurso de diálogo.

Como ponto negativo está o fato de que o usuário precisa conhecer o significado dos termos “entidades”, “relacionamentos” e “atributos” para selecionar o que interessa na consulta nos menus apropriados, pois eles se apresentam em itens separados no menu principal e de forma abreviada (*Ents*, *Rels* e *Properties*, respectivamente). Além disto o usuário deve ter como conhecimento anterior os passos a serem seguidos para realizar a consulta, o que na maior parte das vezes não ocorre. Todo este conhecimento necessário o usuário deve adquirir fora do ambiente da ferramenta, uma vez que em nenhum momento da interação é apresentado o ícone de ajuda no menu.

#### 4.5.2 QBD\*

A QBD\* (*query by diagrams*) é baseada em 4 idéias básicas de seus autores, conforme (Angelaccio et al., 1990 (a)) e (Angelaccio et al., 1990 (b)): 1) representar a parte intencional da base de dados por meio do modelo conceitual ER; 2) empregar um ambiente totalmente gráfico como “interface amigável” com o sistema; 3) definir uma linguagem gráfica padronizada; 4) estender a construção da linguagem gráfica



de forma a representar consultas recursivas. Nesta ferramenta o modelo relacional é representado por meio de diagramas e a consulta é expressa no modelo ER por meio de comandos gráficos da interface. A estrutura geral da consulta é baseada na localização de um conceito único, chamado de *main concept*, que pode ser uma entidade ou um relacionamento. A idéia é tornar este *main concept* uma porta de entrada para uma ou mais sub-consultas, estas sub-consultas expressam possibilidades de navegação entre o *main concept* e outros conceitos do esquema.

Esta ferramenta procura afastar o usuário da necessidade de conhecer os termos da área de Banco de Dados. O processo inicia quando o usuário seleciona um elemento chamado de *main concept* (conceito principal) que determina o ponto inicial, isto é, a raiz da consulta, tornando os conceitos de entidade e relacionamento transparentes. Após a escolha do *main concept* o usuário deve escolher seus conceitos vizinhos, utilizando o esquema apresentado na tela no formato do diagrama ER. O fato de fazer uso da seleção de um conceito principal e depois a seleção dos conceitos vizinhos facilita ao usuário no sentido de não necessitar conhecimentos específicos de Banco de Dados. Este fato demonstra a preocupação em buscar um espelho no mundo real, isto é, uma forma de expressão que se aproxime do conceito que se deseja transmitir ao usuário, sem demandar grande esforço cognitivo.

Quando ocorre de o usuário não ter um conhecimento prévio de banco de dados para entender e ler o diagrama ER apresentado e selecionar os conceitos vizinhos de forma correta, a relação correta é realizada pela ferramenta de forma transparente. Isto é, o usuário não precisa ter conhecimento prévio do modelo relacional, nem do esquema de dados original do domínio apresentado, o que também diminui o esforço cognitivo necessário por parte do usuário.

O usuário necessita de um maior esforço cognitivo quando a lista de atributos dos conceitos selecionados é apresentada em uma janela separada, juntamente com os operadores das consultas. Neste momento o usuário precisa entender a diferença entre atributos e operadores lógicos para selecionar o que é necessário para a formulação da consulta. Outro ponto relevante ocorre quanto aos atributos da entidade *main concept*, que são automaticamente incluídos no resultado da consulta, sem que o usuário tenha sido consultado ou informado. Logo, percebe-se a falta de diálogo entre designer e usuário no sentido de a interface expressar o que está ocorrendo, causando uma ruptura na comunicação usuário-sistema.

O uso de recursividade nesta ferramenta é realizado quando o usuário seleciona um ícone especificando que haverá uma relação cíclica, além da seleção normal dos conceitos. Neste ponto o conhecimento exigido do usuário é bastante específico, uma vez que ele precisa entender o processo de recursão e quando ela ocorre, para poder informar ao sistema. Percebe-se que não houve uma preocupação no sentido de diminuir o esforço do usuário para atravessar os golfos de execução e avaliação, pois nenhum recurso facilitador foi utilizado para amenizar o problema de falta de domínio do usuário quanto ao conceito de recursividade.

#### 4.5.3 Super

O Super é uma ferramenta que faz uso do modelo ERC+, conforme seus autores em (Dennenbouy et al., 1995). Este modelo é baseado no modelo semântico de dados e, além dos operadores da álgebra relacional comum aos modelos de dados relacionais, contém operações que facilitam a construção de consultas aos usuários leigos, exigindo menos conhecimento específico de banco de dados por parte do usuário.

Outro ponto forte do modelo ERC+ é o fato de ele ser baseado no modelo semântico de dados. O modelo semântico de dados permite ao usuário mudar a visão do sistema já definido, onde o usuário tem condições de alterar o diagrama ER apresentado, de forma a facilitar o entendimento e a busca da solução desejada, sem prejudicar o sistema original, pois o processo de mudança ocorre no nível de visão (Gueiber, 2001). Com esta alteração o usuário tem condições de aproximar o sistema de seu mundo real, facilitando a operação e busca do seu objetivo.

Para selecionar as entidades e relacionamentos o esquema da base de dados é apresentado ao usuário no formato de diagrama ER, para o modelo ERC+. Neste momento o usuário necessita de um conhecimento prévio para entender e ler o diagrama ERC+, e então selecionar os elementos de interesse para sua consulta. Com relação aos atributos, os que estão ligados às entidades e relacionamentos selecionados no esquema de dados acompanham o objeto, utilizando o modelo ERC+. Desta forma o usuário consegue visualizar os atributos que ele necessita na consulta já no esquema de dados e seleciona os objetos de seu interesse com mais precisão. Neste ponto está sendo utilizada a técnica WYSIWYG, que entre outros benefícios, diminui a possibilidade de erros, uma vez que o usuário somente poderá fazer sua escolha entre as opções disponíveis.

Após o usuário ter selecionado os elementos interessantes à consulta, o sistema converte o sub-esquema em uma estrutura árvore. Vendo o esquema em forma de uma árvore fica simples para o usuário perceber a hierarquia existente no esquema original, uma vez que esta estrutura é familiar ao usuário, logo a distância semântica é pequena e o esforço cognitivo exigido do usuário é menor.

O modelo ERC+ não suporta uma estrutura cíclica. Quando o usuário, por meio da visão da estrutura em árvore, percebe que a estrutura cíclica está

quebrando a árvore, fica evidente a necessidade de unificar ou desconectar os objetos. Isto ocorre de forma natural sem necessidade de o usuário conhecer os conceitos de estrutura cíclica ou recursão. Novamente a técnica WYSIWYG é utilizada diminuindo a distância semântica e articulatória e facilitando a compreensão por parte do usuário.

Um ponto negativo desta ferramenta é o uso de diversas janelas-sobrepostas, ao invés de utilizar o conceito de janela lado-a-lado. O usuário tem necessidade de adquirir ciência do conteúdo das janelas para dar continuidade ao processo e perceber a conexão entre elas. Este fato gera um esforço cognitivo extra de memorização que seria desnecessário caso fossem utilizadas janelas lado-a-lado, onde o usuário pode ter uma visão do processo como um todo.

## 5. A FERRAMENTA VIQUEN

Conforme visto no capítulo anterior, é importante lembrar que existe no mercado um grande domínio de SGBDs relacionais que, por conseqüência, detêm um vasto legado de informações. Este contexto torna importante que as aplicações de consulta que utilizam linguagens visuais disponibilizem acesso às bases de dados criadas dentro desta abordagem. Dentre o grande número de ambientes de consulta para bases de dados está o VIQUEN, uma ferramenta construída para o modelo ERC+. Este capítulo procura dar uma visão geral da ferramenta VIQUEN, escolhida para o desenvolvimento da interface proposta, e para tanto foi realizada uma avaliação detalhada na ferramenta baseada no método de análise proposto no capítulo 2, que enfoca a visão semiótica do processo de avaliação.

### 5.1 DESCRIÇÃO DA VIQUEN

VIQUEN é um ambiente interativo para consulta visual e extração de esquemas de uma base de dados. Este protótipo foi construído e descrito por Gueiber em (Gueiber, 2001) com o objetivo de atingir a categoria de usuários comuns, os quais buscam freqüentemente por informações armazenadas em sistemas de banco de dados, mas desconhecem as sintaxes de linguagens de consulta, construídas tipicamente para uso por especialistas. Esta aplicação pertence à categoria de editores gráficos tal qual Pasta-3 (Kuntz & Melchert, 1989), QBD\* (Angelaccio et al., 1990 (a)) e SUPER (Dennenbouy et al., 1995), que fazem uso da linguagem visual com a qual o usuário interage para formular suas consultas, sem ser necessário que ele conheça a linguagem textual de manipulação de dados

utilizada pelo SGBD em questão. As operações disponíveis estão implícitas nas ações tomadas pelo usuário sobre o próprio esquema apresentado durante a formulação das consultas, ou seja, a consulta é construída por meio da manipulação direta de objetos do modelo mediante a linguagem visual aplicada sobre ele. O VIQUEN recupera um esquema de base de dados construído sobre o modelo relacional, de pouco poder de expressão do ponto de vista do usuário leigo, e traduz para o modelo ERC+.

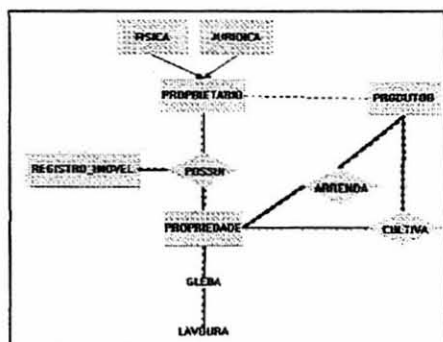
O modelo ERC+ foi utilizado por Gueiber (Gueiber, 2001) no desenvolvimento do VIQUEN como suporte as interações do usuário, ou seja, é valendo-se dos conceitos deste modelo que o usuário formulará suas consultas. Este modelo é uma extensão do modelo entidade-relacionamento, projetada para suportar objetos complexos. Sua meta é aproximar a representação de objetos do mundo real, sendo que esse modelo usa o conceito de tipo entidade para descrever objetos, e o de tipo relacionamento para representar relacionamentos entre objetos, conforme descrito por Spaccapietra et al. em (Spaccapietra & Parent, 1992) e (Spaccapietra et al., 1995).

O protótipo VIQUEN apresenta as seguintes etapas na interface com o usuário, conforme Gueiber em (Gueiber, 2001): a engenharia reversa, a qual traduz uma base relacional para o correspondente modelo ERC+; a tarefa de seleção do sub-esquema desejado na consulta; a especificação do predicado; a visualização da resposta.

Com base na análise do protótipo VIQUEN foi possível descrever as etapas citadas nos elementos que as compõem. No processo de engenharia reversa, o esquema, contendo todas as tabelas e restrições pertencentes ao esquema selecionado pelo usuário, é traduzido automaticamente pela aplicação para um

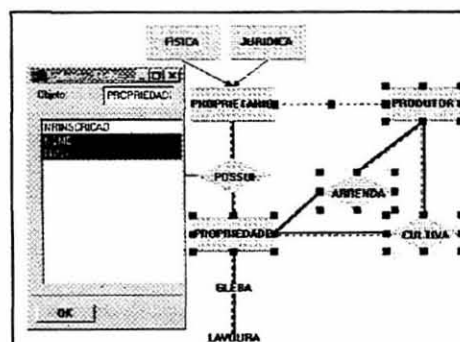
esquema no modelo ERC+. Nesta representação podem ser visualizadas entidades, relacionamentos e atributos complexos que são apresentados graficamente ao usuário em uma janela. As ligações entre os elementos desta janela são feitas através de arestas (figura 9). Em uma outra janela, ativada pelo usuário por meio de uma caixa de diálogo (figura 10), são apresentados os atributos atômicos. O fato de os atributos atômicos não serem exibidos no grafo permite que o esquema não se torne saturado, o que dificultaria a manipulação do modelo exibido.

Figura 9: Esquema de uma propriedade agrícola sob o modelo ERC+



Fonte: Gueiber, 2001

Figura 10: Seleção de atributos sob o modelo ERC+

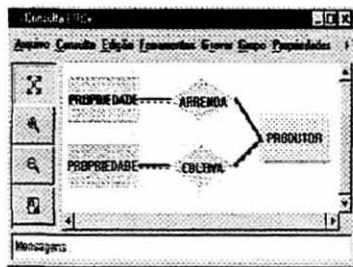


Fonte: Gueiber, 2001

A partir do esquema recuperado, o usuário pode selecionar, valendo-se do mouse, os objetos desejados na formação da consulta. Esta seleção deve incluir tanto os objetos a serem desejados na resposta como aqueles necessários na formação do predicado. Para submeter uma consulta o usuário deve determinar, também, qual visão deseja, escolhendo uma raiz para a consulta. A seleção da raiz da consulta é uma etapa bastante importante na seleção do esquema, pois é esta escolha que irá determinar a hierarquia dos objetos participantes da consulta a ser realizada (figura 11). Em alguns casos, é preciso que o usuário elimine eventuais ciclos que existam entre os objetos da consulta (figura 11). Por último, a fim de

complementar consultas, o usuário pode adicionar subesquemas a ela, estes subesquemas adicionados devem ser unidos ao subesquema inicial, eliminando objetos comuns aos dois subesquemas (figuras 11 e 12).

Figura 11: Exemplo de uma consulta representada por um grafo acíclico com raiz escolhida



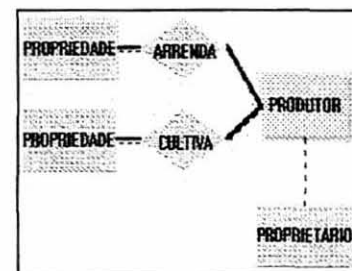
Fonte: Gueiber, 2001

Figura 12: Subesquema proprietário adicionado a consulta



Fonte: Gueiber, 2001

Figura 13: Consulta após a unificação dos subesquemas

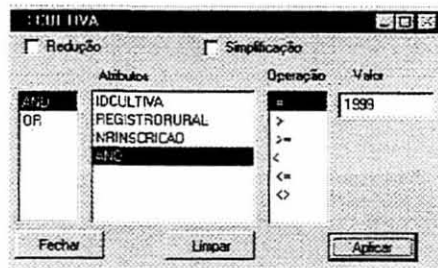


Fonte: Gueiber, 2001

A especificação de predicados é considerada um ponto crítico em consultas a bases de dados, por se tratar de uma tarefa que pode atingir um elevado nível de complexidade, em se tratando de usuários não especialistas. Para realizar esta tarefa o usuário deve selecionar o objeto da consulta, isto é, a entidade desejada, e os atributos que irão compor a resposta da consulta. Na mesma janela, deve ser selecionada a complementação do predicado através da seleção dos operadores lógicos e relacionais e das constantes, para construir expressões booleanas sobre os atributos atômicos (figura 14). Feito isto, serão apresentados ao usuário, em uma nova janela, os predicados estabelecidos por ele, em forma de árvore. A representação visual do predicado em forma de árvore demonstra a hierarquia estabelecida pelo usuário quando selecionou a raiz no esquema de consulta. Também é apresentada, na mesma janela, a instrução SQL que foi gerada automaticamente pela aplicação (figura 15).

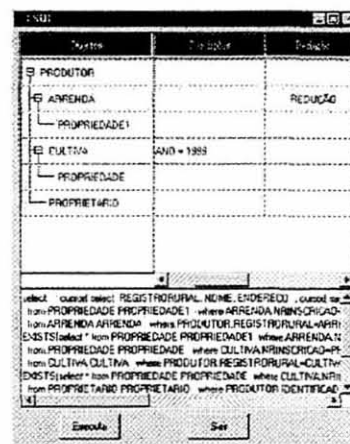


Figura 14: Seleção da complementação do predicado



Fonte: Gueiber, 2001

Figura 15: Representação do predicado em forma de árvore hierárquica e SQL



Fonte: Gueiber, 2001

A última etapa da aplicação é a visualização da resposta, onde os objetos selecionados durante a consulta permanecem igualmente vistos como objetos (figura 16). Esta representação em estrutura não planar, mas de forma hierárquica a partir da raiz, selecionada pelo usuário anteriormente, possibilita que o usuário expanda as instâncias da resposta.

Figura 16: Resposta da consulta mantendo a estrutura de objetos e representação hierárquica em forma de árvore

Objeto	Valor
PRODUTOR	
REGISTRORURAL	9802
NOME	Lucas Oliveira
ENDERECO	Pabão
ARRENDADA	
DATAINICIO	01/01/1999
DURACAO	12
CULTIVA	
ANO	1999
PROPRIEDADE	
NOME	Fazenda Campo Alto
LOCAL	Pabão
PROPRIETARIO	
NOME	Lucas Oliveira
ENDERECO	Mata Alto
CIDADE	Pabão
UF	SP
PRODUTOR	
ENDERECO	
CIDADE	

Fonte: Gueiber, 2001

## 5.2 AVALIAÇÃO DA INTERFACE DA FERRAMENTA VIQUEN

Aproveitando a definição da LEMD na seção 2.5.1 é possível representar o conteúdo da mensagem contida na interface, isto é, o modelo de usabilidade do software. A avaliação realizada na VIQUEN aproveita o potencial da LEMD de representação estruturada da mensagem do designer para determinar se o modelo de usabilidade que chega até o usuário quando este interage com o sistema é condizente com o modelo pretendido pelo designer.









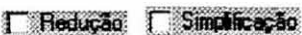
Esta avaliação é motivada pela necessidade de localizar da forma mais precisa possível as partes da interface onde as mensagens do designer não refletem o modelo de usabilidade. Este resultado é importante no processo de redesign da interface do sistema, objeto deste trabalho.

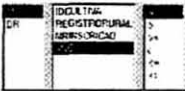
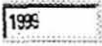
O primeiro passo da análise consistiu em extrair da interface do software VIQUEN elementos com significado aparentemente relevante para o processo, de modo a construir um vocabulário específico da aplicação. Mais especificamente,

tratou-se de levantar os componentes verbais do *domain-register* (Prado & Baranauskas, 1999). Com isto foi possível estabelecer relações do tipo significante-significado entre os elementos presentes no ambiente de trabalho (signos do domínio) com o qual o usuário interage e, na medida do possível, o seu significado para o usuário.

Do ponto de vista semiótico, o que foi feito foi identificar e caracterizar expressões dos signos através das diversas substâncias. Cada uma dessas expressões tem associado um conteúdo (Boscarioli et al. 2001). Assim foi possível a criação de um dicionário de signos, apresentado na tabela 2:

Tabela 2: Dicionário de signos do VIQUEN

Objeto	Significado
	<p>Botão para Seleção de Objetos</p> <p>Botão para Aumento de Visualização (Zoom Maior)</p> <p>Botão para Aumento de Visualização (Zoom Menor)</p> <p>Botão para Movimentação de Objetos</p>
	Objeto do Sub-Esquema: Entidade
	Objeto do Sub-Esquema: Relacionamento
	Objeto do Sub-Esquema: Raiz
	Objeto do Sub-Esquema: Cardinalidade (indicação de item obrigatório)
	Objeto do Sub-Esquema: Cardinalidade (indicação de item não obrigatório)
	Objeto do sub-esquema: especialização/generalização
	Objeto do Sub-Esquema: Atributo Complexo
	Caixas <i>Check-Box</i> para escolha de Operação

	Complementação do predicado: Seleção dos operadores lógicos e relacionais e expressões booleanas sobre os atributos atômicos
	Complementação do predicado: Caixa para escolha das constantes

Fonte: Boscarioli et al. 2001

Complementando o dicionário, foram consideradas as tarefas disponibilizadas pela interface no processo de consulta visual a uma base de dados identificada na seção 4.2 (Boscarioli et al. 2001). A tabela 3 apresenta as tarefas subseqüentes do processo de consulta, bem como suas sub-divisões. Este trabalho compreende somente as etapas 1 e 2, esquema e seleção do sub-esquema, respectivamente. As etapas 3 e 4 estão sendo abordadas em trabalhos relacionados, a etapa de Construção da Consulta está sendo relatada em (Boscarioli, 2001) e a etapa de Visualização do Resultado está sendo descrita em (Ranthum, 2001).

Tabela 3: Tarefas (Etapas) do processo de consulta ao VIQUEN

1. Seleção do Esquema	1.1 Seleção do esquema desejado
2. Seleção do sub-esquema	2.1 Seleção dos objetos que irão compor o sub-esquema
	2.2 Seleção dos atributos relevantes para a consulta
	2.3 Seleção da raiz que determina a visão da consulta
	2.4 Retirada de ciclos
	2.5 Adição de novos sub-esquemas
	2.6 Unificação dos sub-esquemas
3. Construção da consulta	3.1 Seleção dos objetos que comporão o predicado
	3.2 Construção do predicado
	3.3 Apresentação hierárquica do predicado construído
	3.4 Apresentação SQL do predicado construído
4. Visualização do resultado	4.1 Apresentação hierárquica da resposta da consulta construída

Fonte: Boscarioli et al. 2001

### 5.2.1 Avaliação com base nas Medidas Semióticas

Partindo do potencial identificado, na seção 2.5.1, de utilização da LEMD para a análise da comunicação designer-usuário e dando um passo à frente, o método de avaliação proposto foi utilizado em um conjunto de janelas que foram isoladas e trabalhadas com uma certa independência. Esta avaliação da efetividade da mensagem do designer com base na LEMD, como a que foi realizada, complementa os resultados obtidos por métodos de avaliação cognitivos, como aquele apresentado na seção 4.5.

Neste ponto da análise, dispomos da declaração dos signos do domínio da aplicação que estão compondo o dicionário (tabela 2) e estarão presentes na interface para visualização e manipulação pelo usuário e do modelo de funcionalidade que permitem especificar possíveis estruturas de interação que correspondam ao modelo de interação planejado pelo designer, e mesmo representar seqüências completas de passos que o usuário deve executar para atingir uma determinada meta dentro do seu escopo de trabalho (tabela 3). Isto leva à representação LEMD, da etapa de esquema e sub-esquema objeto deste trabalho, apresentada a seguir:

```

Command-Message Esquema // Etapa 1
Combine {
  Command-Message Acesso ao Oracle for Application-function
  Informa_Usuário
  Join {
    View "Nome usuário"
    Enter Information of Nome do Usuário/Esquema
    Select {
      Activate Start Application-Function Informa_Usuário
      Activate Waive Application-Function Informa_Usuário
    }
  }

  Command-Message Acesso ao Oracle for Application-function
  Informa_Senha
  Join {

```

```

    View "Senha"
    Enter Information of Senha
  Select {
    Activate Start Application-Function Informa_Senha
    Activate Waive Application-Function Informa_Senha
  }
}
} // Termina Esquema

Command-Message Consulta ERC+ //Etapa 2
Combine {
  Command-Message Exibe Esquema for Application-function Esquema
  Join {
    View < Esquema Relacional Selecionado Traduzido para o
    Esquema ERC+ - Esquema Original >
    Waiting-for-Input {
      Repeat {
        Sequence {
          Join {
            Select Information-of < Objeto do
            Esquema >
            Select Activate Show Command-
            Message < Seleção de Atributos >
          }
        }
      }
    }
  }
}

Command-Message Seleção Atributos for Application-function Seleção de
Atributos
  Join {
    View "Objeto"
    Show <Nome do Atributo Selecionado>
    Join {
      Sequence {
        Select Information-of "Atributos do Objeto
        Selecionado"
      }
      Activate Start Application-function Seleção de
      Atributos
    }
  }
}

Command-Message Seleção Raiz for Application-function Seleção Raiz
  Join {
    Combine // Seleciona Raiz
    Select Information-of < Esquema_Entidade Relacionamento >
    Sequence {
      Activate Start Application-function Consulta
      Activate Start Application-function Raiz
    }
    Join { // Elimina Ciclos
      Select Information-of <Esquema Ligação>
      Activate Start Application-function Elimina_Ciclos
    }
    Combine { // Adiciona Sub Esquema
      Select Information-of <Esquema Original>
      Select Information-of <Objeto do Esquema>
      Activate Start Application-function Adicionar
      // Unificar Sub Esquema
    }
  }
}

```

```

        Select Information-of <Esquema_Entidade
        Relacionamento em Duplicidade>
        Activate           Start           Application-function
        Unificar Sub Esquema
    }
} // Termina Consulta ERC+

```

Após captar e representar na LEMD o conteúdo da mensagem do designer, é preciso verificar a consistência da mensagem com relação ao modelo de usabilidade planejado por ele. A fonte de obtenção do modelo de usabilidade adotada foi a descrição do protótipo VIQUEN em (Gueiber, 2001), assim como os contatos com a ferramenta. Para melhor localização do leitor iremos utilizar a divisão de etapas constante na tabela 3 baseada na descrição realizada na seção 4.2 deste trabalho, que fez uso da numeração das etapas. Também para fins de avaliação, convencionou-se que o usuário deve ter conhecimento médio no padrão WIMP. A seguir são descritos os pontos críticos identificados.

Na etapa 1, de seleção do esquema desejado, não fica claro se o usuário deve digitar seu nome ou o nome do esquema que deseja editar. Somente com o auxílio do *help* do sistema é que conseguimos entender que o usuário pode ter acesso a um só esquema ou a um conjunto de esquemas ao qual ele estará autorizado pelo administrador de banco de dados.

Na etapa 2, de seleção do sub-esquema desejado para a consulta, podemos analisar cada uma das sub-etapas separadamente. Na sub-etapa 2.1 o usuário é apresentado ao modelo ERC+ sob cuja ótica o esquema é descrito. Nesta janela, o usuário toma contato com os ícones da interface que irão acompanhar todo o processo de consulta e deve compreender seu significado. Estes ícones estão apresentados no dicionário de signos da tabela 2 e são específicos da área de

Banco de Dados, acrescidos de peculiaridades do modelo utilizado (ERC+). Desta forma deduz-se que o designer parte do pressuposto de que o usuário tem domínio destes termos técnicos, ou de que este terá treinamento prévio, pois em nenhum momento existe uma preocupação em apresentá-los ao mesmo. Na sub-etapa 2.2, o processo de seleção dos objetos que devem fazer parte da consulta é realizado de forma natural pelo usuário que deve conhecer interfaces padrão WIMP, visto ser esta uma condição primária para uso da ferramenta. Porém é difícil para o usuário ter ciência de que precisa selecionar objetos neste momento da consulta, pois não existe indicativo algum da necessidade desta seqüência de passos. É natural para usuários com familiaridade com o padrão WIMP que o clique duplo do mouse sobre o objeto deve mostrar características próprias deste, no caso, atributos. Porém, novamente não é apresentada ao usuário a necessidade de ele realizar esta operação neste momento da consulta.

Na seqüência, o usuário deve selecionar um dos objetos que será escolhido como a raiz da consulta (sub-etapa 2.3), na qual a representação hierárquica da resposta será baseada. Esta etapa é de extrema importância para que o usuário tenha êxito em sua consulta, pois é ela que irá determinar a visão da resposta. Logo, o usuário deve ter conhecimento deste fato. A seleção é feita por meio de um clique do mouse sobre o objeto escolhido como raiz, e seleção do item *consulta* no menu de opções, exposto na parte superior da janela, que abre em um sub-menu contendo os itens raiz e predicado. Porém, fica totalmente obscuro na interface o que o item *consulta* do menu significa, e qual o seu impacto, assim como a necessidade de se escolher uma raiz para a consulta nesta fase da interação. A documentação da ferramenta esclarece o que a interface deixa obscuro, ou seja, preenche as lacunas na comunicação interface-usuário.



Após a seleção da raiz, é apresentada ao usuário uma nova janela que contém o sub-esquema selecionado na visão desejada. Neste momento, o usuário poderá deparar-se com um diagrama contendo um ciclo, o que não é natural em um diagrama no modelo ERC+. Porém, novamente é necessário que o usuário tenha algum domínio técnico da área de Banco de Dados para entender que está diante de um problema. A forma de retirar este ciclo é posicionando com o cursor sobre a ligação que deverá ser retirada e pressionando o botão do mouse para eliminar o ciclo e a ferramenta automaticamente duplica o objeto comum da ligação eliminando o ciclo. Para a retirada deste ciclo, seria importante um assistente, pois não fica aparente a necessidade do processo de retirada, até porque é o único ponto da interface onde o usuário pode realizar uma ação sobre um ícone de ligação. Este procedimento, que diz respeito à sub-etapa 2.4, não é encontrado na interface, somente na documentação descritiva da ferramenta.

Ao se deparar com a necessidade de acrescentar novos objetos ao sub-esquema, o usuário estará entrando na sub-etapa 2.5 do processo de construção do sub-esquema de trabalho. Neste caso é natural que ele retorne à janela onde se encontra o esquema original. O que não fica claro na interface é que qualquer nova seleção que o usuário fizer na janela do esquema original será transferida para a janela do sub-esquema onde ele está trabalhando.

Quando a adição de um outro sub-esquema ocorre, o usuário percebe que o novo sub-esquema selecionado não está ligado ao sub-esquema anterior e, naturalmente, tenta fazer esta unificação que indica a sub-etapa 2.6 da construção do sub-esquema. Como nestes casos irão existir objetos duplicados nos dois sub-esquemas, a unificação é feita posicionando o cursor sobre um dos objetos duplicados e pressionando o botão do mouse, quando automaticamente a

ferramenta realiza a unificação. Este passo é considerado natural para um usuário com o domínio necessário da área de Banco de Dados.

Durante a definição das mensagens na LEMD para a etapa 2, em muitos momentos foi necessária a consulta à documentação descritiva da ferramenta. Sem este auxílio seria muito difícil descobrir a seqüência de passos necessária para a construção da LEMD. Isto deixa clara a necessidade de um assistente para orientar o usuário nas etapas do processo de construção e apoiar o usuário na recuperação após os erros. Também é necessário que o usuário compreenda os elementos constituintes de um diagrama no modelo ERC+ para poder reconhecer a existência de um ciclo e a necessidade de retirá-lo, ou mesmo unificar sub-esquemas selecionados.

Neste ponto o sub-esquema está em condições de receber o predicado, que atua como um filtro à base de dados, e decide a resposta a ser apresentada ao usuário. No entanto, o objeto deste trabalho diz respeito somente às etapas de esquema e seleção do sub-esquema.

### 5.2.2 Conclusão da Avaliação

Durante a descrição do dicionário de signos da interface (tabela 2) ficou nítida a necessidade de que o usuário tenha um domínio mínimo da área de Banco de Dados para que consiga interpretar os signos apresentados na interface. O uso de um assistente para ensinar os conceitos próprios do modelo ERC+, orientar os passos que o usuário deve seguir durante a construção do sub-esquema e orientá-lo em algumas situações de erro previstas pela aplicação mostra-se crucial.

O modelo de tarefas – que explicita as idiosincrasias dos ambientes de bancos de dados visuais, se constitui em contribuição para o design deste tipo de

sistemas, no momento em que ele aponta para aspectos críticos no contexto de hipótese e instrumenta o designer com um elenco de pontos que merecem tratamento adicional, seja nas escolhas de *widgets* e imagens a serem feitas para a interface, seja no design das mensagens do designer que deverão compor o sistema de ajuda ao usuário.

Como a interface apresentou lacunas na comunicação com o usuário final, a análise mostra a necessidade de se propor um novo modelo de usabilidade junto ao designer, bem como a necessidade de inclusão de novos signos e componentes de uma linguagem híbrida que possibilite a criação de uma interface que efetivamente expresse toda a meta-comunicação designer-usuário necessária no ambiente de hipótese.

## 6. A INTERFACE PARA A SELEÇÃO DO SUB-ESQUEMA

As análises realizadas nos ambientes visuais de consulta a banco de dados, descritos nos capítulos 4 e 5, revelam que, embora as vantagens destes ambientes visuais sejam amplamente ressaltados na literatura, são necessários, ainda, estudos mais aprofundados capazes de atestar a real facilidade potencial oferecida por este tipo de ambiente. Este fato permitiu a proposta de uma nova interface para o ambiente em questão, a ferramenta VIQUEN. Este capítulo descreve a nova interface proposta baseada na metodologia descrita no capítulo 3, utilizando os conceitos semióticos e cognitivos, além dos recursos de uma linguagem visual sistêmica apresentados no capítulo 2.

### 6.1 CARACTERIZAÇÃO DA INTERFACE

Com base nas análises realizadas nas ferramentas acadêmicas Pasta-3, QBD\* e SUPER e ainda nas avaliações semióticas realizadas no protótipo VIQUEN, todas focando a etapa de geração do sub-esquema da consulta, foi possível visualizar situações favoráveis e desfavoráveis do ponto de vista de interface visando um usuário com pouco conhecimento no domínio de Banco de Dados, mas com conhecimento de interfaces multimodais no ambiente Windows.

Na tentativa de buscar metáforas para o mundo real em um domínio de banco de dados visuais, pode-se concluir que os elementos do modelo (tabela 2) são, por definição, abstratos e devem ser compreendidos para uso adequado. Portanto, no estudo em questão, muitos dos elementos que compõem os sub-ambientes fazem uso de termos técnicos ao invés de ícones que representem uma metáfora do

mundo real. Esta escolha muitas vezes é necessária, em função de que o uso de um ícone irá demandar um esforço cognitivo muito maior por parte do usuário do que a utilização do termo técnico correspondente. Esta situação é bastante comum quando o domínio da aplicação tem jargão técnico bastante difundido na comunidade na qual se insere. Para minimizar o esforço do usuário leigo em tomar conhecimento destes termos, uma solução adequada é o uso de *helps* no sistema, que auxiliem na caracterização dos conceitos técnicos do domínio de aplicação (no caso, o modelo ERC+). Outra opção é a utilização de exemplos da própria base de dados em questão, fazendo uso das instâncias, isto é, dos valores dos atributos, dos quais o usuário, mesmo leigo, tenha conhecimento.

Devido à existência de diversas etapas que o usuário deve seguir para poder ter êxito na montagem da consulta desejada, conforme demonstra a tabela 3, é de suma importância a necessidade de um assistente para orientar o usuário nas etapas do processo de construção e apoiar o usuário na recuperação após os erros. Sem este auxílio seria muito difícil descobrir a seqüência de passos necessária para construção da consulta, principalmente para um usuário leigo.

A idéia é que o assistente inicie o processo requisitando o nome do esquema original de consulta, utilizando uma caixa de diálogo onde o usuário poderá escolher o diretório e os arquivos de esquema existentes neste diretório por meio de caixa de seleção (figura 18). Também será disponibilizada ao usuário a utilização de sub-esquemas de consultas já realizadas que foram gravadas para utilização posterior.

Com relação à seleção da raiz da consulta, a idéia é trabalhar com a metáfora de uma sentença. Uma sentença, em sua forma mais simples, é composta por sujeito e predicado. Levando em consideração que na gramática o sujeito é a parte fundamental da sentença onde todo o resto se refere a ele (verbo e objeto), a idéia é

que a raiz da consulta fará o papel do sujeito da sentença. Para tanto, a proposta é de que, por meio de assistente e diálogo entre sistema e usuário, a primeira seleção a ser feita no esquema original seja a da raiz da consulta, onde, baseado na metáfora, o sujeito é o ponto de partida da construção da sentença. Neste ponto está sendo aproveitada a proposta da ferramenta QBD\* (seção 4.5.2), onde a seleção da raiz como conceito principal da consulta é a primeira a ser realizada. Após a seleção da raiz da consulta a cor do objeto selecionado como raiz será diferenciado (figura 20). A partir daí o sub-esquema poderá ser mostrado no formato de uma sentença, onde a raiz da consulta é o sujeito da sentença, utilizando a metáfora proposta (figura 21).

Para a seleção dos objetos do esquema original que deverão fazer parte da consulta, gerando o sub-esquema e continuando o processo de construção da sentença, o usuário deverá fazer a seleção no próprio diagrama e uma janela auxiliar irá mostrar os objetos que já foram selecionados (figura 22). Valendo-se desta janela o usuário poderá selecionar os objetos um-a-um, de forma completa, isto é, aproveitando todo o esquema, ou retirando objetos já selecionados. O recurso da cor será utilizado no momento de selecionar os objetos, sendo que o esquema original será apresentado ao usuário na cor preta e os objetos que o usuário for selecionando com o mouse na cor azul claro, diferenciando-os. Como o sistema computacional oferece o recurso de seleção automática dos objetos não selecionados pelo usuário, mas que são necessários para manter o caminho e integridade do modelo, estes objetos serão selecionados automaticamente, mas não por vontade do usuário e sim necessidade do modelo subjacente. Como foi dito no parágrafo anterior, o recurso da cor também será utilizado para identificar a raiz da consulta, onde o objeto selecionado pelo usuário para ocupar o lugar de raiz da

consulta será marcado pela cor laranja. Após a seleção dos objetos que comporão o sub-esquema, este será mostrado em forma de uma sentença com seus objetos na cor preta (figura 23).

Também é necessário que o usuário compreenda os elementos constituintes de um diagrama no modelo ERC+ para poder reconhecer a existência de um ciclo e a necessidade de retirá-lo, ou mesmo unificar sub-esquemas selecionados. Logo, um assistente aliado a um diálogo entre o sistema e o usuário poderá guiar o usuário no sentido de “enxergar” um ciclo que deverá ser retirado ou mesmo a necessidade de buscar novos objetos no esquema original. Aliado a isto e com base na ferramenta SUPER (seção 4.5.3), que faz uso do fato de que visualmente um ciclo em um diagrama representa uma situação anormal, a apresentação do sub-esquema em forma de sentença, que é um formato conhecido pelo usuário, facilitará a localização de um ciclo, que é algo anormal nesta estrutura também (figuras 23 e 24). Para visualizar a necessidade de outros objetos, lacunas na construção da sentença de consulta poderão identificar este fato, além da necessidade de atributos relevantes para a resposta da consulta ou sua construção que não estarão presentes no sub-esquema selecionado (figuras 25, 26 e 27).

A partir daí e também com base na experiência do SUPER, para que o usuário perceba a necessidade de unificar os novos objetos selecionados ao sub-esquema de trabalho serão mantidos os novos objetos selecionados em cor azul escuro, de forma a diferenciá-los daqueles que já compunham o sub-esquema de trabalho, que estarão em cor preta (figuras 28, 29 e 30).

Na seqüência, o próximo passo refere-se à seleção dos atributos dos objetos que farão parte da consulta e está baseado na interface da ferramenta VIQUEN (seção 5.1). Neste ponto o usuário será guiado por um assistente para selecionar os

objetos um a um a partir do diagrama, o objeto selecionado ficará na cor azul claro e a partir daí o usuário poderá selecionar os atributos deste objeto (figura 32). Também o recurso da cor será utilizado, quando o usuário for selecionando os atributos do objeto, que ficarão na cor azul claro. Na janela auxiliar o usuário poderá visualizar os atributos, referentes ao objeto marcado que já foram selecionados ou, até mesmo, remover algum atributo selecionado, da mesma forma que ocorre com a seleção dos objetos do sub-esquema (figura 33). No caso específico de um objeto conter mais de 15 atributos, somente os 15 primeiros serão mostrados no diagrama; os demais estarão relacionados na janela auxiliar que irá conter um painel com todos os atributos do objeto em questão, apresentando do lado esquerdo os atributos que compõem o objeto e do lado direito os atributos do objeto que já foram selecionados pelo usuário. Isto está baseado na ferramenta Pasta-3 (seção 4.5.1) e tem como objetivo diminuir a quantidade de itens do diagrama, deixando-o mais livre para facilitar a visualização por parte do usuário.

Também um recurso a ser utilizado, buscando uma melhor interação usuário-sistema, é o de dica (*hint*), conhecido do usuário no ambiente Windows onde, quando o usuário permanecer com o cursor por algum tempo sobre um objeto o sistema mostrará uma instância deste objeto existente na base de dados que está sendo utilizada. Como partiu-se do princípio de que o usuário conhece o domínio da base de dados apresentada, a visualização de um conteúdo real da instância poderá ajudá-lo na escolha deste objeto como parte integrante ou não do sub-esquema. O mesmo recurso será utilizado quando da escolha dos atributos dos objetos. Isto se deve ao fato de que, muitas vezes, o nome dado ao objeto ou atributo pelo administrador de banco de dados ou analista do sistema não condiz com o nome



que o usuário está acostumado a utilizar, logo visualizando a instância, ficará mais fácil de o usuário reconhecer o objeto e transportá-lo para o mundo real.

Levando-se em consideração Norman que cita em (Norman, 1986) que os sistemas computacionais devem ampliar as capacidades dos usuários, e que o usuário tende a se tornar perito no uso do software, os auxílios serão acionados pelo usuário por meio de um botão *Ajuda* no rodapé da tela e o usuário terá possibilidade de utilizar o menu para avançar nas etapas e sub-etapas da construção da consulta. Também será possível ao usuário desfazer o que foi realizado retornando à etapa anterior, através de um botão Etapa Anterior, também no rodapé da tela. O recurso de *hint*, citado anteriormente, também estará disponível em todos os botões da interface.

## 6.2 A METODOLOGIA PARA A ESPECIFICAÇÃO DA INTERFACE

De acordo com a ES, o processo de design consiste em estruturar uma mensagem a ser direcionada ao usuário, que transmite o modo de utilização do sistema e o significado dos elementos com os quais o usuário está interagindo. É conveniente ao designer poder representar explicitamente esta mensagem durante o processo de construção da interface, para que a mesma seja estruturada e coerente.

A nova interface para uso em consulta a banco de dados visuais na ferramenta VIQUEN, no que diz respeito à seleção do sub-esquema da consulta, seguirá a proposta descrita na seção anterior. Fazendo uma análise do perfil do usuário e das tarefas que este usuário desempenha no processo de consulta a um banco de dados visuais (seção 4.2) e após a análise da ferramenta VIQUEN utilizando a proposta de Prado e Baranauskas, citado na seção 5.2.1, e suas conclusões citadas na seção 5.2.2, além dos estudos realizados nas ferramentas

Pasta3, QBD\* e SUPER, citadas na seção 4.5, foi possível reprojeter a interface. Assim foi concebido o seu modelo de usabilidade, descrito por meio da LEMD. Como a todo conteúdo corresponde uma expressão, como por exemplo *widgets* e elementos visuais, utilizou-se a STAG para representar a realização da funcionalidade no ambiente. Assim ambos, LEMD e STAG, constituem a descrição da interface proposta. A confecção de auxílios e manuais também pode ser considerada parte deste processo, derivando tanto diretamente do designer, quanto do modelo de usabilidade por ele estruturado e da própria interface já esboçada, orientada pelo domínio escolhido que exige um assistente e auxílios que orientem o usuário relativamente aos conceitos técnicos da área de Banco de Dados.

### 6.2.1 Análise do Perfil do Usuário

Parte-se do princípio que o usuário deste tipo de ferramenta tem conhecimento do domínio da consulta a ser realizada, isto é, tem uma meta bem definida quanto aos resultados desejados a partir da consulta construída. Como a ferramenta de interação será desenvolvida em um aplicativo voltado para ambiente Windows, por ser este o ambiente operacional mais difundido em computadores de pequeno porte, também parte-se do princípio que o usuário domina o ambiente multimodal de interface Windows, bem como suas características principais.

### 6.2.2 Análise de Tarefas

Baseado no domínio escolhido para o presente trabalho, seleção do sub-esquema para consulta a base de dados no modelo ERC+, foram relatadas as seguintes tarefas do usuário:

1. SELEÇÃO DO ESQUEMA
  - 1.1 Digitar o nome do usuário
  - 1.2 Digitar a senha do usuário
  - 1.3 Selecionar o esquema desejado para a consulta dentre os permitidos
2. SELEÇÃO DO SUB-ESQUEMA
  - 2.1 Selecionar o objeto desejado como raiz da consulta
  - 2.2 Selecionar o(s) outro(s) objeto(s) que comporão a consulta
  - 2.3 Eliminar ciclos (ligações circulares) existentes no sub-esquema formado
    - 2.3.1 Selecionar a ligação que deverá ser retirada para eliminar o ciclo
  - 2.4 Adicionar novos elementos ao sub-esquema formado
    - 2.4.1 Adicionar novos objetos
      - 2.4.1.1 Selecionar novo(s) objeto(s) do esquema original apresentado
    - 2.4.2 Adicionar um novo sub-esquema já gravado
      - 2.4.2.1 Selecionar novo sub-esquema já gravado apresentado
    - 2.4.3 Unificar os novos elementos adicionados ao sub-esquema formado
      - 2.4.4.1 Selecionar o objeto que está em duplicidade nos dois sub-esquemas
  - 2.5 Selecionar atributos dos objetos que compõem o sub-esquema
    - 2.5.1 Selecionar objeto
    - 2.5.2 Selecionar atributos do objeto

## 6.3 A APLICAÇÃO

### 6.3.1 As Etapas da Aplicação

A aplicação inicia o processo de construção do sub-esquema da consulta requisitando o nome do usuário e sua senha, conforme figura 18. Embora a validação do usuário e da senha pareça semanticamente deslocada, ela se justifica aqui pois a partir das informações do usuário é possível detectar os esquemas aos quais o usuário tem permissão de acesso. Desta forma, será apresentada ao usuário uma caixa de diálogo onde poderá ser escolhido o esquema por meio de caixa de

seleção. Também será disponibilizada ao usuário a utilização de sub-esquemas de consultas já realizadas que foram gravadas para utilização posterior.

Pressionando o botão **Recuperar Esquema**, o esquema selecionado será apresentado na tela fazendo-se uso do modelo ERC+, onde os objetos que compõem o esquema estarão todos na cor preta, conforme figura 19. Neste momento o usuário deverá selecionar a raiz da consulta, clicando com o mouse sobre o objeto que representa a entidade que ele deseja como tema da consulta. O objeto selecionado ficará marcado mudando sua cor para laranja, mostrado na figura 20. Caso o usuário não tenha familiaridade com a ferramenta, em qualquer momento do processo de construção da consulta, por meio do botão **Ajuda** ele poderá ter conhecimento sobre quais tarefas estão disponíveis no estágio corrente, gerando um auxílio contextual, e o procedimento necessário para solicitar sua execução. Para encerrar esta fase o usuário deverá pressionar o botão **Selecionar Raiz**, quando será aberta uma nova janela, referente ao sub-esquema de trabalho, onde o esquema será apresentado ao usuário na forma de uma sentença utilizando a raiz selecionada como sujeito, fazendo uso então da metáfora escolhida e propondo ao usuário a construção de uma consulta utilizando a raiz selecionada como tema, mostrado na figura 21.

O próximo passo será a seleção dos outros objetos que irão compor a consulta. Caso seja necessário o esquema inteiro para que a consulta seja bem sucedida, o usuário deverá selecionar todo o esquema, ou apenas objetos isolados que sejam de interesse. Novamente o recurso da cor, neste caso o azul claro, é utilizado para mostrar ao usuário que o objeto foi selecionado. O usuário realiza a seleção dos objetos no próprio esquema, posicionando com o cursor no objeto

desejado e pressionando o botão do mouse, mostrado na figura 22. Na mesma figura é possível visualizar uma janela auxiliar onde o usuário realiza a aceitação da seleção, nesta janela um quadro apresenta os objetos já selecionados e apresenta duas opções: **Selecionar Todo** o esquema e **Retirar** a seleção realizada por engano. Ao terminar a seleção de todos os objetos desejados no sub-esquema de consulta, o usuário deve pressionar o botão **Confirmar** na janela auxiliar. Para diminuir a necessidade de retorno, ao pressionar o botão **Confirmar** o sistema apresenta uma mensagem perguntando se existe mais algum objeto a ser selecionado. Desta forma o usuário será induzido a pensar na resposta desejada da consulta e verificar se todos os objetos desejados estão sendo considerados. Caso o usuário responda que não existem mais objetos a serem selecionados o comando do sistema passa para a janela de Elimina Ciclos.

Como o usuário poderá selecionar objetos isolados que, de alguma forma, necessitem de outros objetos não selecionados para que o caminho e integridade do modelo sejam mantidos, estes objetos serão selecionados automaticamente pelo sistema. Então será apresentada uma nova sentença, sobrepondo-se à anterior na mesma janela, contendo somente os objetos selecionados pelo usuário e os selecionados automaticamente pelo sistema, mantendo a raiz da consulta como sujeito da sentença, conforme mostrado na figura 23. Nesta etapa o sub-esquema apresentado voltará a ser da cor preta.

No próximo passo o usuário deverá reconhecer a eventual existência de um ciclo e a necessidade de retirá-lo. Neste ponto da construção do sub-esquema o botão **Ajuda** irá indicar como realizar a tarefa. Para desfazer um ciclo o usuário deve posicionar com o cursor na ligação que está gerando o círculo na sentença e

pressionar o botão do mouse, utilizando o recurso da cor a ligação selecionada ficará azul claro, mostrado na figura 24. A partir da seleção da ligação e pressionar o botão **Eliminar Ciclo**, automaticamente o sistema irá duplicar o objeto que está gerando o ciclo, abrindo o círculo, conforme figura 25. Neste caso a metáfora da sentença que está sendo construída será composta de uma sentença interna. Ao terminar a verificação de ciclos, o usuário deve pressionar o botão **Próximo Passo**, indicando que esta etapa foi encerrada e os ciclos foram retirados. Caso não haja ciclos a serem retirados o usuário passa para a próxima sub-etapa pressionando o botão **Próximo Passo**. A qualquer momento, quando o usuário pressionar o botão **Próximo Passo** o sistema apresenta uma mensagem perguntando se existe algum ciclo na sentença apresentada, mostrado na figura 24. Desta forma o usuário será induzido a visualizar alguma estrutura circular que não é normal em uma sentença.

A seguir será apresentada ao usuário uma nova sentença sem as estruturas cíclicas (figura 25) e onde o usuário poderá adicionar outros objetos que, porventura, tenham sido esquecidos. Para tanto basta pressionar o botão **Objetos** e o sistema apresentará a janela referente ao esquema original com os objetos já selecionados em azul escuro (figura 26), onde o usuário poderá proceder à seleção dos objetos adicionais da mesma forma que no início do processo. Logo após a confirmação da seleção dos novos objetos (figura 27), o sistema passará à janela de Unificação dos Novos Elementos apresentando o sub-esquema de trabalho e os novos objetos escolhidos, conforme figura 29. Este processo está explicado no texto ao que se tem acesso pela pressão do botão **Ajuda**. Em caso de o usuário ter entrado por engano na tela de Seleção de Novos Objetos ele poderá através do botão **Próximo Passo** passar para a tela de Adição de Novos Elementos.

Também é possível o usuário adicionar um outro sub-esquema já gravado nesta nova consulta, para tanto basta pressionar o botão **Sub-Esquema** na janela de Adição de Novos Elementos (figura 25). Neste momento o sistema apresenta uma nova janela onde o usuário poderá selecionar o nome do arquivo onde foi gravado o sub-esquema desejado (figura 28). Porém somente serão apresentados sub-esquemas que pertençam à mesma base de dados do esquema original selecionado no início da construção da consulta, evitando erros por parte do usuário. Para executar a tarefa solicitada o usuário deverá pressionar o botão **Novo Sub-Esquema**, quando o sistema passará para a janela de Unificação dos Novos Elementos apresentando o sub-esquema de trabalho e o novo sub-esquema escolhido. Caso o usuário tenha entrado por engano na tela de Seleção de Novo Sub-Esquema ele poderá retornar para a tela de Adição de Novos Elementos pressionando o botão Próxima Etapa.

Então, os novos objetos e/ou sub-esquemas selecionados estarão na cor azul escuro, e neste momento o usuário deverá proceder à unificação dos sub-esquemas, uma vez que ele irá se deparar com os sub-esquemas desconectados na tela e com cores diferentes, mostrado na figura 29. Para tanto basta posicionar com o cursor no objeto que está em duplicidade nos dois sub-esquemas e pressionar o botão do mouse, o recurso da cor irá mudar a cor do objeto selecionado para azul claro, conforme figura 30. Ao pressionar o botão **Unificar Sub-Esquema**, o sistema, automaticamente, procede à união dos sub-esquemas. Neste momento o sub-esquema de consulta volta a ter a cor preta original (figura 31). Após a unificação dos sub-esquemas e/ou objetos, o sistema apresenta uma mensagem perguntando se existe mais algum sub-esquema a ser unificado, mostrado na figura 30. Desta

forma o usuário será induzido a pensar se todos os objetos estão integrados em um só sub-esquema.

A partir do momento em que o usuário tiver o sub-esquema selecionado é que será feita a seleção dos atributos dos objetos que serão relevantes para a consulta, como componentes da resposta ou somente como restritivos na composição da consulta. Neste ponto o usuário deverá selecionar os objetos um-a-um. Como, a princípio, nenhum atributo dos objetos é considerado para a consulta, o usuário tem obrigação de realizar a seleção, sob pena de realizar uma consulta sem atributos. Quando o usuário selecionar um objeto por meio do mouse, o objeto ficará marcado na cor azul (figura 32) e a partir da pressão do botão **Selecionar Atributos** uma nova janela será aberta. A partir daí, o usuário irá selecionar os atributos do objeto marcado e realizar a aceitação da seleção na janela auxiliar, onde são apresentadas duas opções, além da seleção um-a-um: **Selecionar Todos** os atributos do objeto marcado em azul e **Retirar** a seleção de algum atributo realizada por engano. A medida em que o usuário seleciona atributos no sub-esquema, estes são relacionados em um quadro na janela auxiliar, que pode ser visualizado na figura 33. Para que a tarefa selecionada seja realizada é necessário pressionar o botão **Confirmar**, que fecha a janela auxiliar e permite que outro objeto seja selecionado e procede-se da mesma maneira para selecionar atributos para os outros objetos. Ao terminar a seleção dos atributos de todos os objetos do sub-esquema deve-se pressionar o botão **Próximo Passo**.

No caso específico de um objeto conter mais de 15 atributos, somente os 15 primeiros serão mostrados no sub-esquema os demais estarão relacionados na



janela auxiliar que irá conter um painel com todos os atributos do objeto em questão, porém não selecionados.

A partir deste momento o usuário poderá passar para a próxima fase da consulta, que é a construção do predicado, etapa a ser abordada em outro trabalho.

As sub-tarefas componentes da tarefa corrente serão disponibilizadas juntamente com a mensagem correspondente, parte do programa em LEMD, que tem por função explicitar para o usuário o estado do sistema e o que este espera do usuário. No entanto, sempre que o usuário necessitar de ajuda com relação à seqüência de passos para a construção da consulta, poderá utilizar o botão **Ajuda** que indicará procedimentos específicos para o estágio de interação corrente. Também o recurso de *hint* estará disponível no aplicativo, onde o usuário terá condições de compreender o que cada ícone simboliza.

O botão **Cancelar** irá permitir ao usuário cancelar a sub-etapa atual e retornar à sub-etapa anterior do processo, tendo oportunidade de refazê-la. Isto é, caso o usuário esteja selecionando atributos, ao pressionar o botão **Cancelar**, irá retornar à sub-etapa que Adiciona Novos Elementos ao sub-esquema de trabalho; caso o usuário esteja iniciando o processo de construção de predicado da consulta, ao pressionar o botão **Cancelar** irá retornar à sub-etapa de Selecionar Atributos.

Todas as telas apresentam o botão **Etapa Anterior** que permite ao usuário recuperar a etapa imediatamente anterior cancelando o que foi realizado na etapa corrente. Por exemplo, ao iniciar o processo de visualização do resultado o usuário poderá, através do botão **Etapa Anterior** retornar ao processo de construção do predicado.

Para sair do sistema o usuário deverá utilizar o item de menu Arquivo, sub-menu Sair. Neste caso não será armazenado nada do que o usuário fez até o momento. Para poder armazenar o sub-esquema selecionado o usuário deverá finalizar o processo de consulta, isto é, somente após ter selecionado o sub-esquema e construído o predicado é que o usuário terá disponível a opção de Salvar o novo sub-esquema gerado, mas esta etapa fará parte de outro trabalho. No mesmo item de menu Arquivo o sub-item Fechar, irá proceder ao fechamento da consulta corrente sem efetivar as seleções do usuário, oferecendo a possibilidade de selecionar outro esquema para uma nova consulta.

Também no menu existe um item chamado Ajuda, que apresentará ao usuário uma ajuda generalizada envolvendo o ambiente Windows, a ferramenta Viquen como um todo, os conceitos básicos de Banco de Dados e características do ambiente da interface. Desta forma o usuário terá condições de aprender com a ferramenta e aumentar seu nível de interação com ela.

O menu apresenta as três etapas de construção de uma consulta, onde cada item contém suas respectivas sub-etapas. Assim, caso o usuário tenha recuperado um sub-esquema já gravado, poderá selecionar direto a etapa onde pretende realizar uma alteração por meio dos itens de menu. No caso de um usuário mais experiente, ele poderá selecionar a seqüência de sub-etapas sozinho sem passar pelas mensagens de assistente dos botões de cada janela. No entanto, a seqüência de passos para a construção da consulta deverá ser mantida, isto é, sem passar pela seleção da raiz da consulta o sistema não permitirá acesso à seleção de objetos ao sub-esquema, assim como, sem passar pela construção do sub-esquema o sistema não permitirá acesso à construção do predicado da consulta.

### 6.3.2 As Mensagens da Aplicação

Um ponto importante observado nas análises das ferramentas QBD\*, SUPER, Pasta-3 e no próprio VIQUEN foi o fato de não haver uma seqüência definida de tarefas a serem executadas, não dando ao usuário leigo condições de saber qual deve ser a próxima tarefa a ser executada. Desta forma, a aplicação proposta trabalha com um assistente com o intuito de mostrar ao usuário somente a sub- etapa possível de execução naquele momento, por meio de telas sobrepostas. Este assistente apresenta em todas as telas uma linha de cabeçalho de tela informando a etapa e sub-etapa de referência no processo de construção da consulta e no corpo da tela duas linhas, uma informando o que está sendo visualizado no momento, identificando a situação do processo de consulta, e outra linha informando a tarefa que deve ser executada. A tabela 4 apresenta os textos existentes em cada janela da aplicação. Isto é feito para diminuir a necessidade de retorno em caso de erro ou falta de alguma seleção importante para o fechamento da consulta desejada.

Tabela 4: Textos das telas

Etapa	Linha de cabeçalho da tela	Linha do corpo da tela
ESQUEMA	Bem Vindo ao Sistema (figura 18)	<p>Digitar os dados de identificação do usuário.</p> <p>-----</p> <p>Selecionar o esquema desejado para a consulta.</p>
SUB-ESQUEMA	Seleção da Raiz da Consulta (figura 19)	Selecionar o objeto que será a raiz da consulta, funcionando como o sujeito da sentença a ser criada.
	Seleção dos Objetos (figura 22)	<p>Esquema tema da consulta que vai funcionar como o sujeito da sentença construída.</p> <p>-----</p> <p>Selecionar os objetos que irão compor a consulta, funcionando como o predicado da sentença construída.</p>

		Os objetos selecionados farão parte do novo sub-esquema. ----- Selecionar os objetos desejados no esquema original.
	Elimina Ciclos (figura 23)	Sub-esquema selecionado funcionando como a sentença da construída. ----- Eliminar as ligações que estão formando um círculo na sentença construída.
	Adiciona Novos Elementos (figura 25)	Sub-esquema selecionado sem ciclos, funcionando como a sentença construída. ----- Selecionar novos objetos ou novo sub-esquema se desejado.
	Seleção de Novo Sub-Esquema (figura 28)	Selecionar um novo sub-esquema já gravado para adicionar à nova consulta a ser construída.
	Seleção de Novos Objetos (figura 27)	Esquema tema da consulta que vai funcionar como o sujeito da sentença construída. ----- Selecionar os novos objetos, além dos já selecionados, que formarão o predicado da sentença construída. ----- Os objetos selecionados farão parte do novo sub-esquema. ----- Selecionar os objetos desejados no esquema original.
	Unificação dos Novos Elementos (figura 29)	Sub-esquema selecionado com os novos objetos e/ou sub-esquema. ----- Unificar os objetos e/ou sub-esquema para conectar todos em um só sub-esquema.
	Seleção de Atributos (figura 33)	Sub-esquema selecionado funcionando como a sentença construída. ----- Selecionar o objeto para a seleção de seus atributos. Os atributos selecionados farão parte da consulta ----- Selecionar os atributos desejados no sub-esquema.
<b>PREDICADO</b>	<b>Etapa de Construção do Predicado (figura 34)</b>	

Com o intuito de diminuir a carga cognitiva do usuário leigo sem se descuidar do usuário experiente, a aplicação apresenta um assistente, presente em cada janela no botão **Ajuda** indicando os passos a serem realizados para executar a tarefa requerida na sub-etapa em questão. A tabela 5 apresenta o texto de ajuda existente em cada tela da aplicação. As mensagens foram escritas de forma a seguir um padrão sintático e compor uma linguagem de ajuda sistêmica.

Tabela 5: Texto de ajuda

Etapa	Sub-Etapa	Texto de Ajuda
ESQUEMA	Bem Vindo ao Sistema	Digite seu nome de usuário no sistema e sua senha no sistema. Selecione o diretório onde se encontra o arquivo do esquema desejado e o nome do arquivo do sistema desejado. Depois pressione o botão Recuperar Esquema.
SUB-ESQUEMA	Seleção da Raiz da Consulta	Selecione no esquema original um objeto que será utilizado como sujeito da sentença a ser construída, isto é, a raiz da consulta. Depois pressione o botão Selecionar Raiz.
	Seleção dos Objetos	Selecione no esquema original os objetos desejados para compor o predicado da sentença a ser construída, isto é, o sub-esquema de trabalho. Depois pressione o botão Selecionar Objetos.
		Selecione no esquema original o objeto desejado. Para todos os objetos pressione o botão Selecionar Todos. Para retirar um objeto, selecione o objeto no esquema e depois pressione o botão Retirar. Ao final pressione o botão Confirmar.

	Elimina Ciclos	Selecione no sub-esquema a ligação que está gerando um círculo e depois pressione o botão Eliminar Ciclo. Repita a operação até que o sub-esquema não apresente mais ciclos. Caso não existam mais ciclos a serem eliminados pressione o botão Próximo Passo.
	Adiciona Novos Elementos	Pressione o botão referente ao novo elemento a ser adicionado, Objeto ou Sub-Esquema. Caso não deseje adicionar novos elementos pressione o botão Próximo Passo.
	Seleção de Novo Sub-Esquema	Selecione o nome do arquivo do novo sub-esquema que será adicionado ao sub-esquema atual de trabalho. Depois pressione o botão Novo Sub-Esquema. Caso não deseje adicionar novo sub-esquema pressione o botão Próximo Passo.
	Seleção de Novos Objetos	Selecione no esquema original os novos objetos desejados para compor o predicado da sentença a ser construída. Depois pressione o botão Selecionar Objetos. Caso não deseje adicionar objetos pressione o botão Próximo Passo.
	Unificação dos Novos Elementos	Selecione no sub-esquema o objeto em duplicidade nos sub-esquemas, pressione o botão Unificar Sub-Esquema. Repita a operação até que todos os sub-esquemas e objetos estejam unificados. Caso não existam mais sub-esquemas a serem unificados pressione o botão Próximo Passo.
	Seleção de Atributos	Selecione no sub-esquema o objeto cujos atributos deseja incluir na consulta, pressione o botão Selecionar Atributos. Repita a operação até que todos os atributos tenham sido selecionados. Ao final pressione o botão Próximo Passo.

		Selecione no sub-esquema os atributos desejados. Para todos os atributos do objeto pressione o botão Selecionar Todos. Para retirar um atributo, selecione o atributo no sub-esquema e pressione o botão Retirar. Ao final pressione o botão Confirmar.
--	--	---

Outro recurso utilizado para diminuir o re-trabalho é o uso de mensagens de alerta em pontos críticos do processo, como por exemplo, eliminação de ciclos e unificação de novos objetos e/ou sub-esquemas adicionados. Caso o usuário não realize a eliminação de ciclo a consulta não poderá ser terminada, uma vez que o SGBD que realiza o processamento das informações na base de dados tem como limitação o fato de não tratar recursividade, que estaria representada pela ligação em círculo no diagrama ERC+. Estas mensagens de alerta são apresentadas quando o usuário encerra a tarefa e manifesta a intenção de passar para a próxima sub-etapa da construção do sub-esquema. A tabela 6 apresenta o texto das mensagens apresentadas no encerramento das sub-etapas críticas.

Tabela 6: Texto de mensagens de sub-etapas críticas

Sub-Etapa	Texto da Mensagem
Seleção dos Objetos (figura 22)	Deseja selecionar mais algum objeto?
Elimina Ciclos (figura 24)	Deseja eliminar mais algum ciclo?
Unificação de Novos Elementos (figura 30)	Deseja unificar mais algum sub-esquema?

Também o ambiente Windows oferece o recurso de *hint* onde qualquer ícone apresentado na tela pode ser explicado, bastando que o usuário permaneça com o cursor sobre o ícone por um tempo. Aproveitando o ambiente Windows a aplicação faz uso da manipulação direta por meio do uso do mouse. Neste ambiente os termos

“selecionar”, “posicionar”, “pressionar”, “digitar” já estão difundidos entre os usuários, experientes ou não, que utilizam a tecnologia do computador.

Partindo do princípio que a aplicação trabalha com o recurso WYSIWIG, a probabilidade de ocorrência de erro é mínima, uma vez que o usuário somente poderá manipular elementos existentes na tela e a presença dos mesmos na tela determina que estes podem ser manipulados pelo usuário. Também o uso de poucos recursos de intenção do usuário (selecionar, posicionar, pressionar e digitar) permite uma interação simples e linear, mantendo uma linguagem visual sistêmica. Neste sentido toda a tarefa executada pelo usuário deve ser confirmada por meio de botões específicos de cada sub-etapa, tais como: **Recuperar Esquema**, **Eliminar Ciclo**, **Próximo Passo**, **Confirmar** entre outros. Além disto, a linguagem visual utilizada definiu a ordem e utilização dos botões de rodapé das janelas, a construção das sentenças de ajuda, as linhas de cabeçalho das telas e de corpo das mesmas.

Uma situação importante em qualquer aplicativo é a possibilidade de o usuário poder refazer o que foi realizado, de forma total ou parcial. Para tanto a aplicação disponibiliza o recurso de **Cancelar** a tarefa executada, retorno à **Etapa Anterior**, **Fechar** a consulta que está sendo construída com a possibilidade de abrir outra e **Retirar** a seleção no caso de uma seleção errada de objeto ou atributo.

No menu o usuário tem possibilidade de acessar o item **Ajuda** contendo os sub-itens: **Sobre o Windows** que apresenta um texto padrão do ambiente Windows, explicando seus recursos e como utilizá-los, **Sobre o Viquen** que apresenta um texto informando sobre o ambiente Viquen, seus recursos baseado no modelo ERC+ e uma descrição das etapas de construção de uma consulta neste ambiente, **Sobre Banco de Dados** onde o usuário encontrará conceitos específicos do domínio de



Banco de Dados, incluindo as definições e descrições da simbologia utilizada, tais como entidade, relacionamento, cardinalidade, entre outros, e Sobre o Ambiente onde são descritas as características da interface e os auxílios.

### 6.3.3 O Auxílio ao Usuário

Foi utilizada neste protótipo a documentação extensa *online* onde, por meio de mensagens do designer para o usuário, é possível que o usuário se localize nas diversas etapas do sistema e em relação a qual deve ser sua ação no ponto em que se encontra. Também o uso do auxílio sensível ao contexto por meio de caixas de diálogo que o usuário aciona quando achar conveniente, permite que o usuário tome conhecimento de qual deve ser sua ação para executar a tarefa desejada. O uso de *hint* permite ao usuário tomar conhecimento da ação a ser realizada pelo botão ou ícone, antes de acioná-lo. Como o protótipo foi desenvolvido levando-se em consideração que o usuário tenha conhecimento prévio do ambiente Windows, não houve preocupação em trabalhar mensagens de ajuda para seleção de objetos na interface.

Buscando solucionar as dúvidas típicas do usuário referentes ao uso da aplicação, o protótipo buscou responder às perguntas da seguinte maneira:

- A exploração de metas foi solucionada por meio do menu de tarefas disponíveis no cabeçalho de cada janela.
- O uso de duas frases em cada janela identifica onde o usuário está situado em relação à aplicação e qual deve ser sua tarefa na respectiva janela. Isto procura solucionar o problema de definição e descrição.
- Com o objetivo de amenizar o esforço no sentido de como o usuário deve realizar a tarefa desejada, um botão **Ajuda** em cada janela

permite que seja possível conhecer o procedimento necessário; também os *hints* disponíveis nos ícones e botões permitem ao usuário conhecer a funcionalidade do respectivo botão ou ícone.

- Com o intuito de resolver o problema de diagnóstico o protótipo procura trabalhar como assistente, apresentando ao usuário uma tela após a outra, obedecendo à seqüência de passos necessários para a construção da consulta, e, sempre que o usuário passa para a próxima sub-etapa da construção do sub-esquema, um novo diagrama do sub-esquema é apresentado mostrando as alterações realizadas pela execução da tarefa correspondente. Também o uso de cores no diagrama procura mostrar ao usuário qual foi o objeto selecionado. Desta forma, o usuário tem a percepção do que ocorreu e consegue avaliar se seu objetivo foi atingido. Caso o objetivo não tenha sido atingido, os botões de **Cancelar** e **Etapa Anterior** permitem retornar e refazer o procedimento. A existência de caixas de diálogo em sub-etapas consideradas críticas procura chamar a atenção do usuário na tarefa a ser executada para minimizar a possibilidade de erro futuro. No caso do menu somente estarão disponíveis ao usuário as etapas e sub-etapas de forma a obedecer os passos da construção do sub-esquema da consulta. Assim, se procurou diminuir a possibilidade de erros sempre chamando a atenção do usuário para que ele avalie cada sub-etapa no momento de sua execução, e não mais tarde quando o erro será mais difícil de ser consertado.

- Para o caso de identificação de estado, o nome da tela identifica a etapa e sub-etapa da aplicação em que o usuário se encontra. Também a mudança de cores, utilizada quando o usuário executa uma tarefa selecionando um objeto no diagrama, procura identificar que ocorreu uma

alteração na aplicação. No caso do menu, a medida em que o usuário avança nas etapas e sub-etapas de construção da consulta, as tarefas mudam seu estado valendo-se da tonalidade de cinza indicando se a tarefa está disponível ou não.

#### 6.3.4 A LEMD da interface proposta

O uso da LEMD para definir a interface proposta facilitou o processo de design uma vez que foi possível definir e especificar a mensagem da interface de forma estruturada e abstrata sem a preocupação com os objetos de interface convencionais, isto focou a tarefa e permitiu um aprofundamento no sentido de buscar solucionar as lacunas da ferramenta apresentadas na seção 5.2.

Também o fato de se ter a oportunidade de realizar a análise na ferramenta utilizando a LEMD, que é capaz demonstrar as lacunas da interface no sentido de determinar se o modelo de usabilidade que chega até o usuário quando este interage com o sistema é condizente com o modelo pretendido pelo designer, favoreceu o trabalho no sentido de continuar a definição da nova interface utilizando o mesmo recurso para determinar o novo modelo de usabilidade com o qual o usuário terá contato durante a utilização do software. A LEMD da interface proposta está descrita no apêndice 1, assim como as regras de mapeamento semântico utilizadas na LEMD.

#### 6.3.5 A STAG da interface proposta

Após terem sido descritas as funções da aplicação, definidas pelo modelo funcional da interface através da LEMD, foi necessário definir as ações que controlam e compõem os comandos das funções da aplicação. Para tanto a escolha

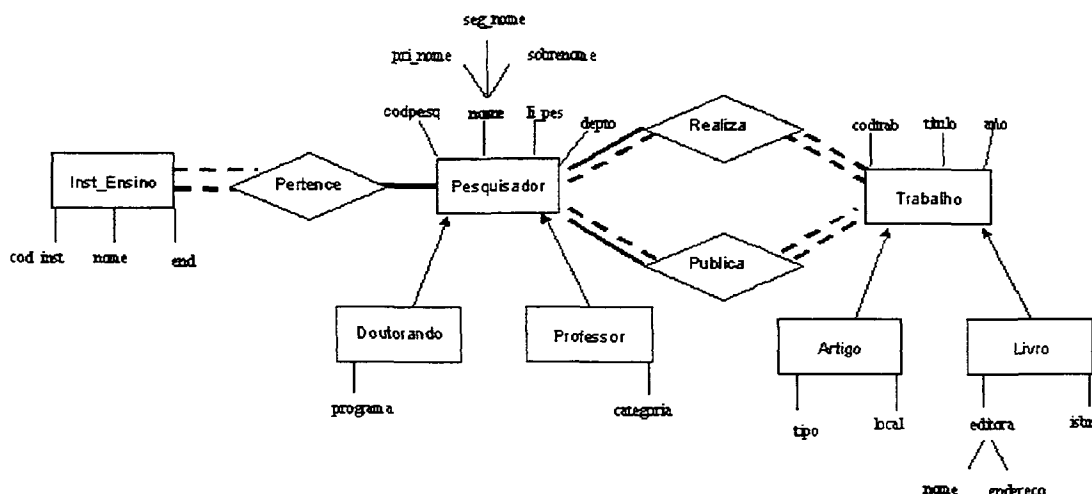
da STAG foi benéfica uma vez que permitiu o mapeamento da tarefa-ação de forma minuciosa pois o método abrange desde a definição os tipos e instâncias do sistema, operações disponíveis, tipo de expressão e conteúdo do sistema e suas instâncias, tipos de expressão e conteúdo do usuário e suas instâncias, mapeamento dos tipos e instâncias do sistema e do usuário, até as regras gramaticais para o usuário realizar suas intenções junto ao aplicativo. No apêndice 2 está sendo apresentada a STAG da interface proposta.

Na seqüência deu-se prosseguimento ao desenvolvimento da prototipação da nova interface utilizando a ferramenta Borland Delphi 5, que permitiu mostrar a interação de acordo com o modelo especificado. A prototipação tem por característica a geração de um produto parcial, semi-acabado que exige pouco esforço de programação e sendo não operacional. A opção pelo uso de prototipagem deu-se pelo fato de que esta reduz consideravelmente o tempo de desenvolvimento e a necessidade de manutenção do sistema, uma vez que o foco do trabalho se ateve no processo de análise de interfaces existentes e na proposta de solução das lacunas de comunicação encontradas nestas interfaces.

#### 6.3.6 O Domínio da Aplicação

O domínio escolhido para aplicação do protótipo foi o de um sistema fictício de controle de publicações da associação de professores universitários do estado do Paraná, conforme diagrama ERC+ da figura 17, baseado no domínio relatado por Rodacki em (Rodacki, 2000).

Figura 17: Diagrama ERC+ do domínio da aplicação



Nesta figura podem ser visualizados diversos conceitos do modelo ERC+, como a entidade Pesquisador que está ligada à entidade Trabalho pelo relacionamento Realiza, que possui a seguinte semântica: “um pesquisador pode realizar zero ou mais trabalhos”, e, “um trabalho pode ser realizado por um ou mais pesquisadores”.

A entidade Pesquisador é uma generalização das entidades Doutorando e Professor, onde a flecha formada por uma linha contínua indica uma generalização “is-a”, que dita que todos os doutorandos são pesquisadores. A generalização indicada por uma flecha formada por uma linha tracejada indica uma generalização “may-be-a” que diz que os professores podem ou não ser pesquisadores. Por exemplo, um professor adjunto, pode não exercer uma atividade de pesquisa.

Ainda na entidade Pesquisador, pode-se verificar a existência de um atributo complexo, nome, que possui atributos obrigatórios, `pri_nome` e `sobrenome`, e um atributo opcional `seg_nome`.

A entidade Trabalho é uma generalização das entidades Artigo e Livro, onde todos os artigos e livros são considerados trabalhos (generalizações “is-a”).

Os atributos `cód_inst`, `codpesq` e `codtrab` são mostrados sublinhados por serem atributos chaves das entidades `Inst_Ensino`, `Pesquisador` e `Trabalho`, respectivamente.

### 6.3.7 Exemplo de Consulta

Na seqüência é apresentada uma consulta realizada sobre o domínio em questão que procura enfatizar itens relevantes do processo da consulta, tal como a seleção de uma raiz da consulta, a seleção de um sub-esquema para a realização da consulta, entre outros.

Quais pesquisadores da UNICAMP ou da UFPR publicaram e/ou realizaram livros e artigos entre os anos de 1995 e 2000?

Considerações:

Metáfora da sentença : Os pesquisadores da UNICAMP ou da UFPR que publicaram e/ou realizaram livros e artigos entre os anos de 1995 e 2000 foram ...

Raiz: Pesquisador

Necessidade de eliminar ciclo do diagrama

Possibilidade de adicionar o objeto “artigo” após a seleção dos objetos que irão compor a consulta

Consulta grande

Redução

Nas figuras seguintes (figuras de 17 a 33) segue a seqüência de janelas da aplicação para realizar a consulta em questão:

Figura 18: Tela inicial da interface.

Figura 19: Apresentação do Esquema Original no modelo ERC+

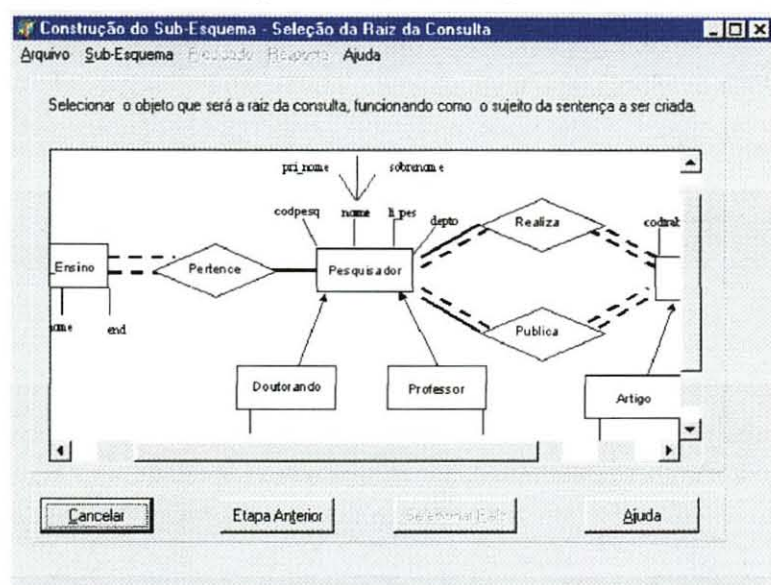


Figura 20: Seleção da raiz da consulta

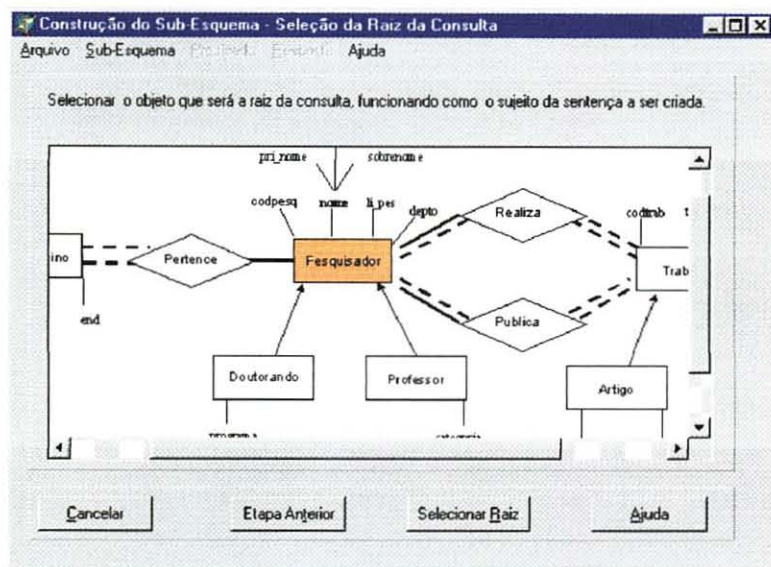


Figura 21: Esquema original utilizando a metáfora do sujeito da sentença para a raiz da consulta

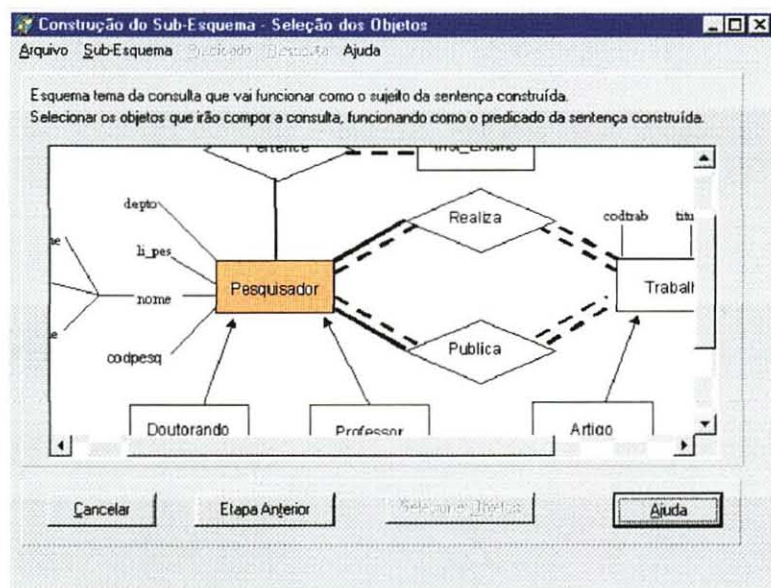




Figura 22: Seleção/remoção dos objetos que comporão o predicado da sentença, que está sendo utilizada como metáfora da consulta.

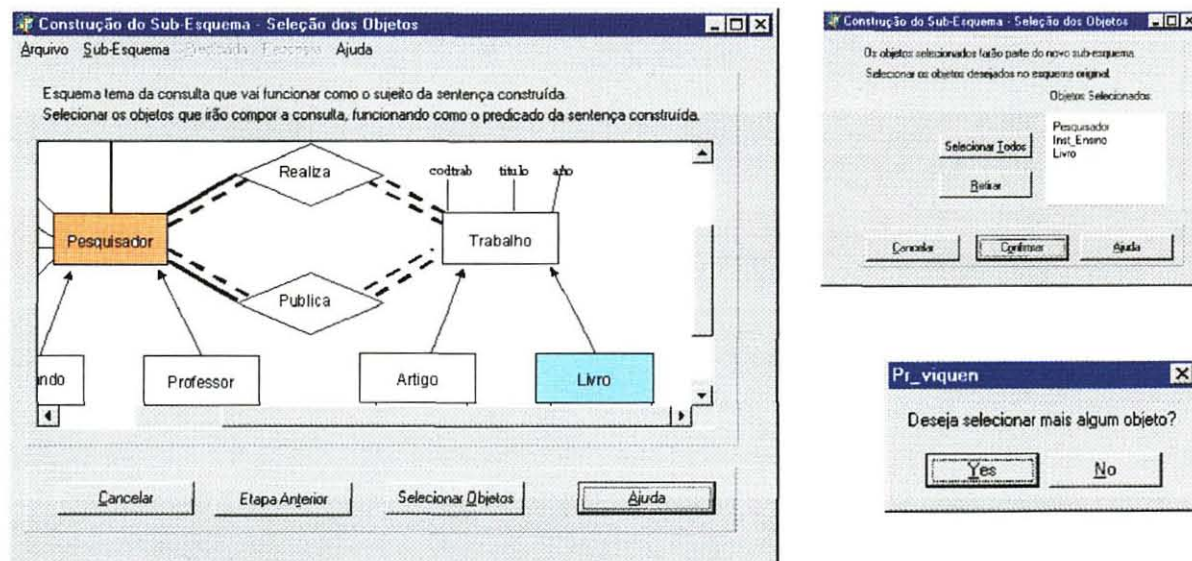


Figura 23: Apresentação do sub-esquema construído, para eliminação de uma estrutura cíclica – Pesquisador/Realiza/Trabalho/Publica/Trabalho.

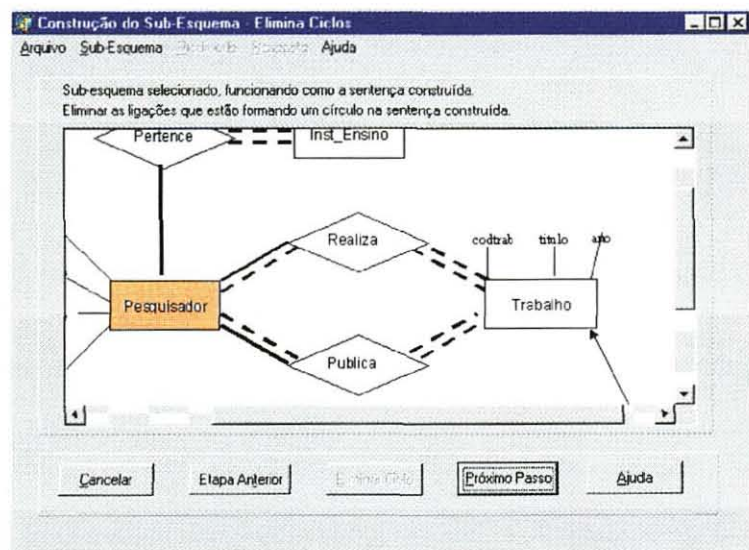


Figura 24: Apresentação da seleção de uma ligação para a abertura da estrutura cíclica.

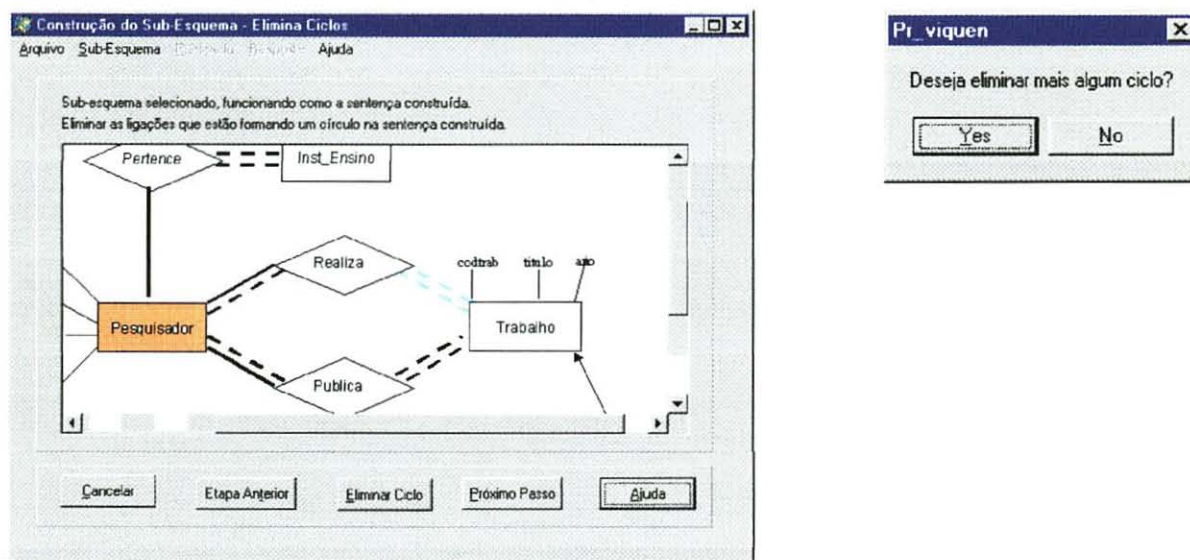


Figura 25: Apresentação do sub-esquema sem ciclos para adição de novos elementos.

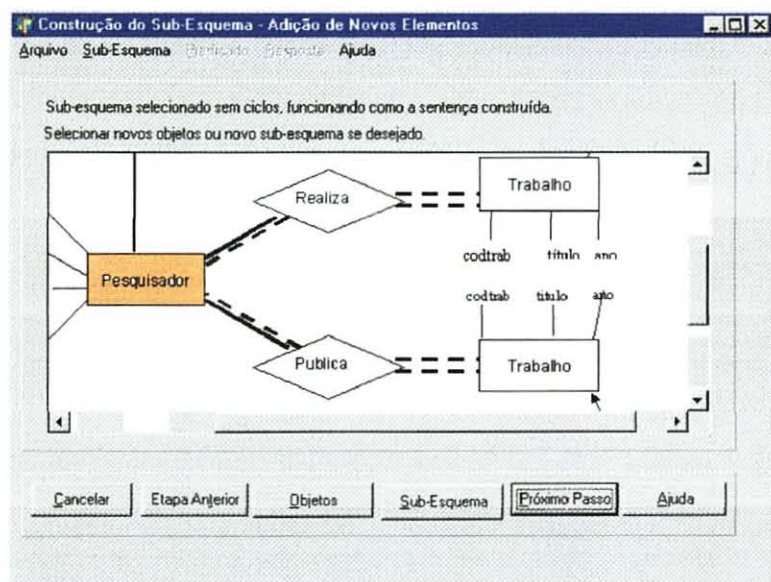


Figura 26: Apresentação da tela para seleção de novos objetos, onde está sendo mostrado o esquema original com os objetos já selecionados para a seleção de novos objetos.

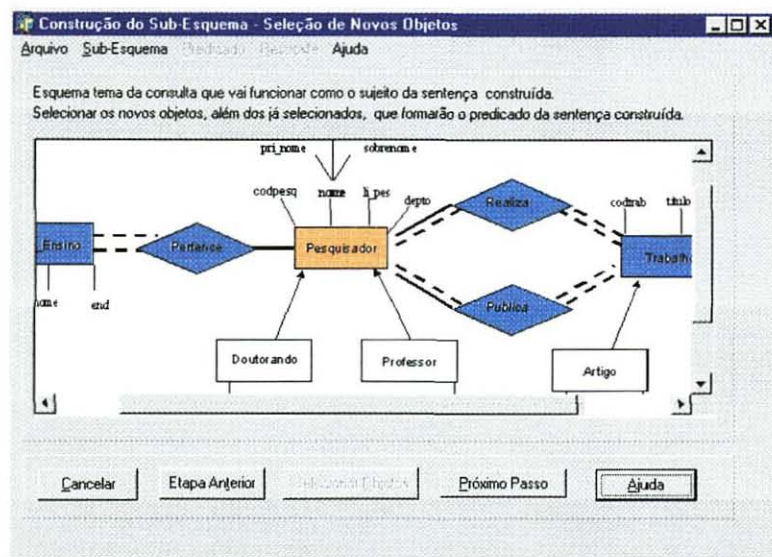


Figura 27: Apresentação da tela mostrando a seleção de um novo objeto, além dos já selecionados anteriormente.

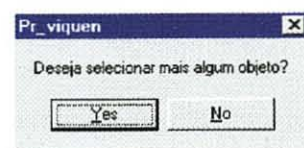
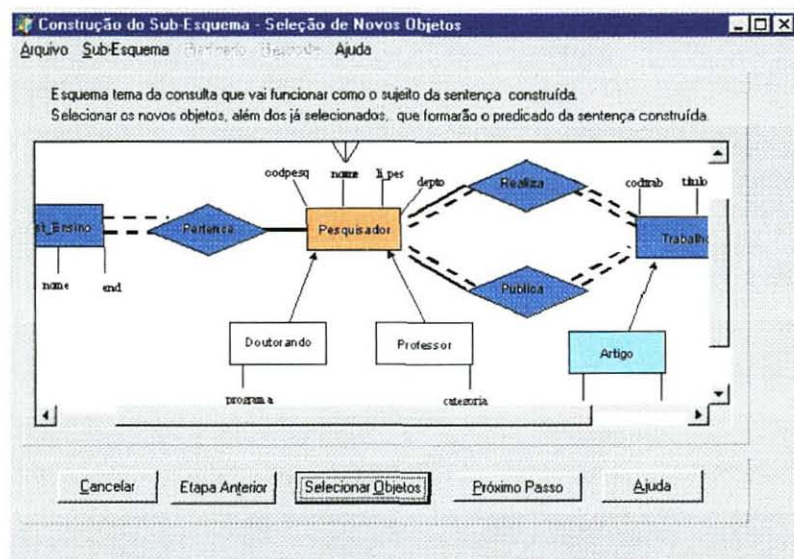




Figura 28: Apresentação de tela mostrando a opção de adição de um novo sub-esquema onde serão apresentados os arquivos já gravados de sub-esquemas utilizando a mesma base de dados do esquema original em uso.

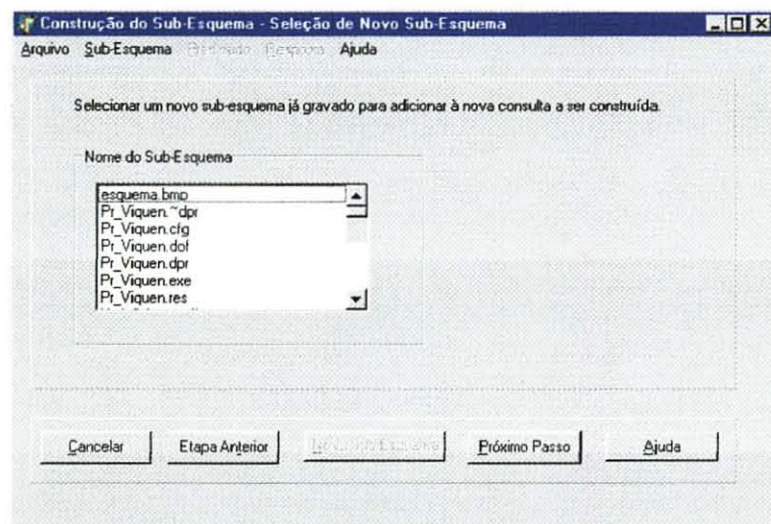


Figura 29: Apresentação de tela mostrando o sub-esquema já selecionado, junto ao novo objeto selecionado para unificação.

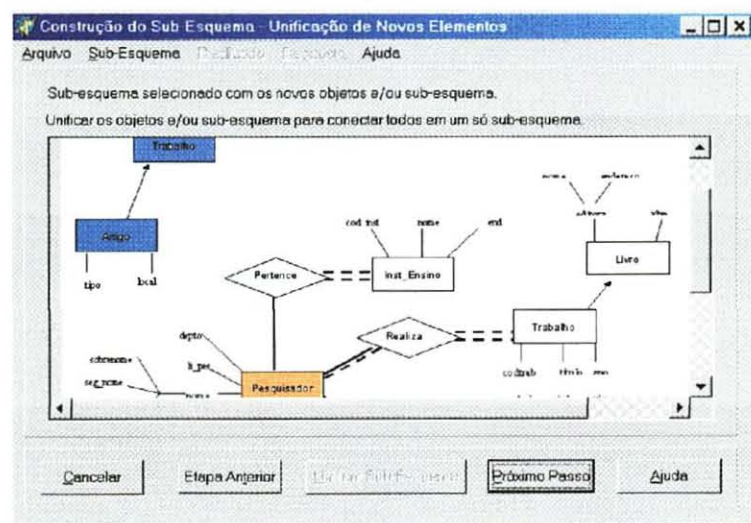


Figura 30: Apresentação de tela mostrando a seleção de um objeto duplicado nos dois sub-esquemas, onde será feita a união.

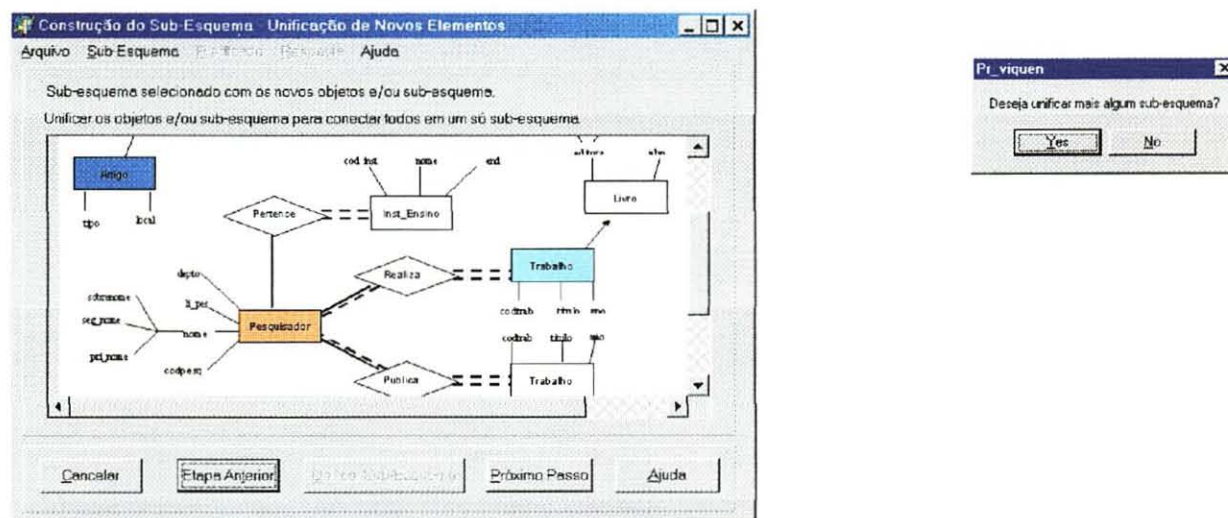


Figura 31: Apresentação da tela mostrando o sub-esquema unificado para seleção dos atributos que comporão a consulta.

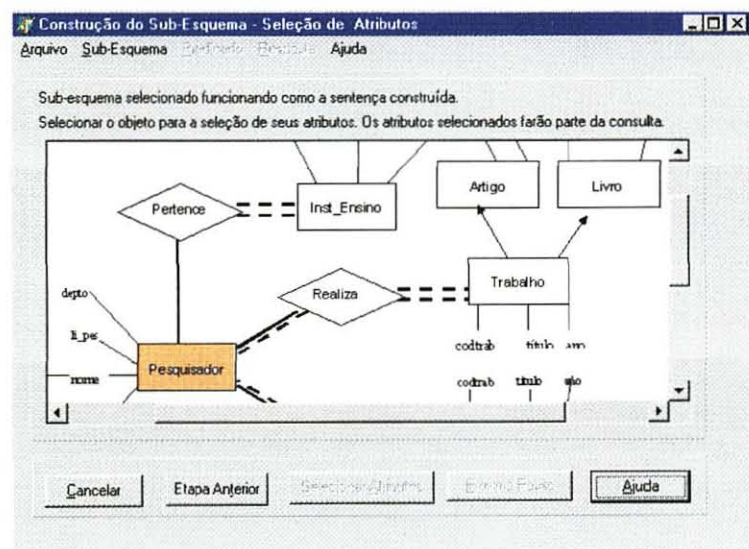


Figura 32: Apresentação de tela mostrando a seleção de um objeto para inclusão de seus atributos na consulta.

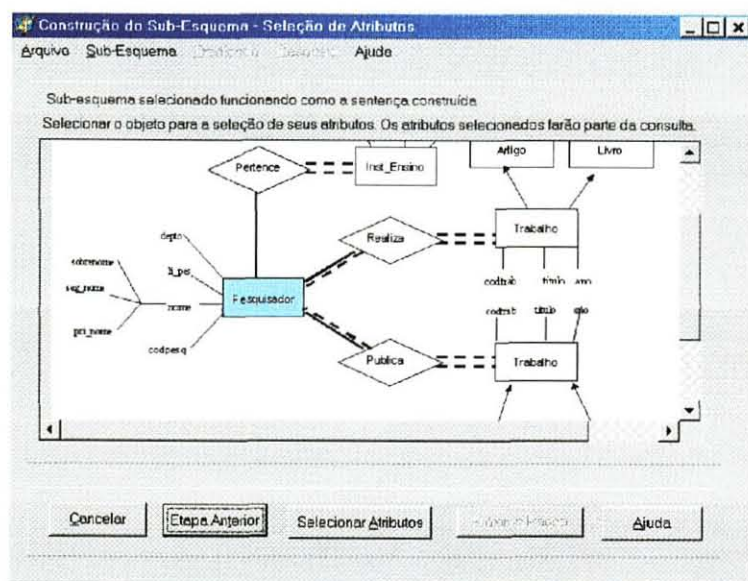


Figura 33: Apresentação de tela mostrando a seleção dos atributos do objeto selecionado.

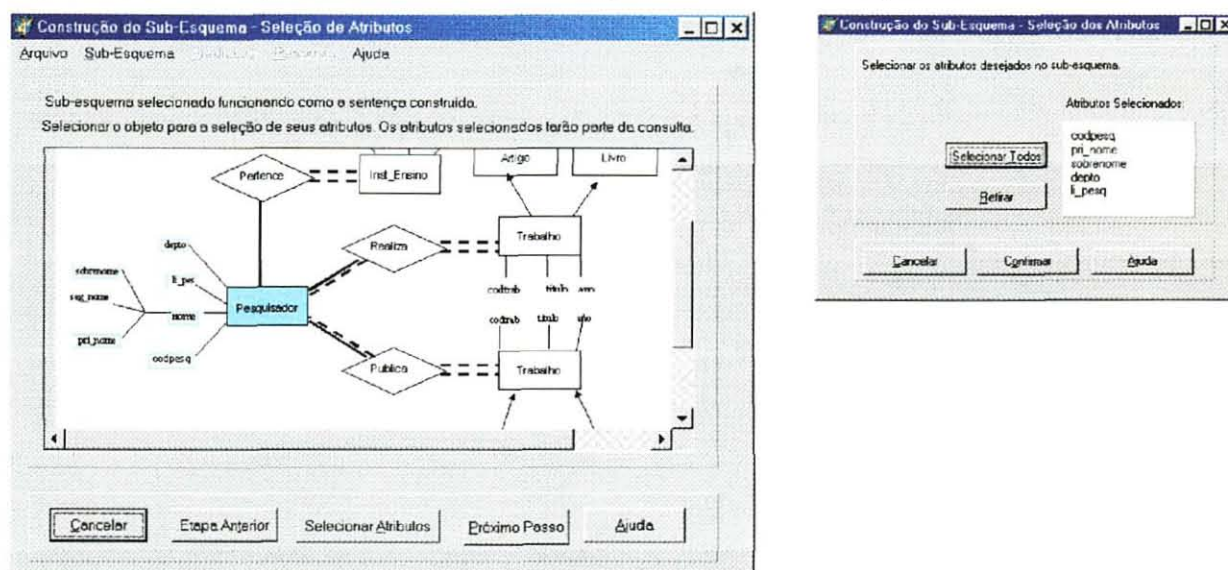
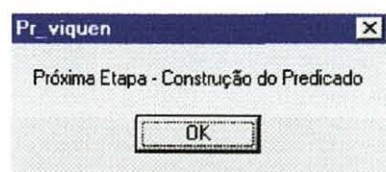


Figura 34: Apresentação de tela finalizando a etapa de construção do sub-esquema, indicando a passagem para outra etapa do processo de construção da consulta.



### 6.3.8 O Auxílio *Online*

Tanto a LEMD como a STAG se mostraram ferramentas eficazes no sentido de nortear o design da interface de forma estruturada e dentro de uma linguagem visual sistêmica e poderiam por si só representar a documentação da interface na forma de auxílio ao usuário. Porém, dentro da proposta de se desenvolver uma interface usável para usuários com pouco conhecimento da área de Banco de Dados é importante que, além do uso de símbolos de expressão que demandem pouco esforço cognitivo, tais como conceitos técnicos, o usuário tenha disponível um conjunto de auxílios para dirigi-lo em caso de dúvida ou erro.

Neste sentido, e também procurando preencher uma das lacunas encontradas na maioria dos aplicativos analisados, foi utilizado o recurso de assistente embutido na interface, desta forma a interface controla a seqüência de etapas e sub-etapas que devem ser atravessadas para que o usuário alcance seu objetivo sem perder a ordem das tarefas necessárias para tal. O usuário tem acesso à ordem de tarefas utilizando os textos nas janelas que indicam em qual sub-etapa ele se encontra e qual a tarefa que deve ser desempenhada naquele momento da aplicação. Este assistente embutido também procura minimizar alguns possíveis erros de seleção fazendo uso de mensagens, onde após cada sub-etapa finalizada o usuário é levado a pensar sobre a ação tomada de forma a avaliar se realmente irá satisfazer ao seu objetivo final.

Para o caso de usuários principiantes ou que não utilizem a ferramenta com freqüência o auxílio disponível é o botão Ajuda que aparece em todas as janelas da interface e dispõe de um texto explicando as ações que o usuário deve tomar para realizar a tarefa proposta.



Já para os usuários com nenhum ou pouco conhecimento na área de Banco de Dados a interface disponibiliza no menu principal o item Ajuda que contém os sub-itens Sobre o Windows, Sobre o VIQUEN, Sobre Banco de Dados e Sobre o Ambiente. No sub-item Sobre o Windows está disponível um texto padrão do Windows onde é possível o usuário se familiarizar com os conceitos deste ambiente de interface. No sub-item Sobre o VIQUEN o usuário irá encontrar um texto sobre a ferramenta, suas características e recursos disponíveis. O sub-item Sobre Banco de Dados tem como objetivo esclarecer ao usuário os conceitos e termos técnicos mais comuns da área e que venham a auxiliar o usuário no sentido de entender o processo de construção da consulta. No sub-item Sobre o Ambiente o usuário tem condições de visualizar a seqüência de etapas e sub-etapas da construção da consulta e um auxílio *online* estrutural da interface mostrando a ordem das janelas seus objetivos, localização dos botões e boxes e forma de preenchimento e seleção dos objetos, assim como execução da tarefa desejada. Este auxílio está descrito no apêndice 3.

### 6.3.9 Comparação com a Ferramenta VIQUEN

Tomando como base os resultados da análise realizada na ferramenta VQUEN, que está relatada na seção 5.2, e a proposta da nova interface para o ambiente VIQUEN, relatado na seção corrente, é possível salientar que a nova interface se propõe a amenizar as lacunas encontradas na análise no que se refere à necessidade do usuário leigo em conhecer alguns conceitos básicos da área de Banco de Dados. Assim a nova proposta procura trabalhar de forma intensa a comunicação designer-usuário através de mensagens na tela e caixas de diálogo, disponibilizando vários níveis de ajuda: assistente de construção da consulta, ajuda



para o usuário se localizar em relação as etapas e sub-etapas do processo de construção da consulta, ajuda para executar as ações solicitadas pelo sistema, ajuda para o usuário conhecer alguns conceitos da área de Banco de Dados e do modelo ERC+ através da ferramenta VIQUEN.

Houve uma especial atenção na formulação da linguagem visual de consulta onde o usuário trabalha com poucas intenções e a formulação das intenções está baseada em um conjunto simples e pequeno de ações oriundas do ambiente Windows, que são conhecidas do usuário. Na ferramenta VIQUEN percebe-se que não houve uma preocupação com a linguagem visual de consulta, uma vez que o usuário realiza suas intenções através de um variado conjunto de ações, tais como, seleção no menu para ativar as tarefas, pressão de botões para edição e, em outros casos, seleção no diagrama.

A nova interface para a ferramenta VIQUEN utiliza os conceitos de banco de dados no que se refere ao diagrama do esquema apresentado na tela, uma vez que, conforme citado na seção 6.1, o uso de uma metáfora para estes conceitos iria demandar um esforço cognitivo maior por parte do usuário do que este conhecer os conceitos que já fazem parte do jargão técnico da área. Para amenizar este esforço além da comunicação designer-usuário, a existência dos nomes dos atributos junto aos elementos do diagrama e o uso do *hint* para indicar a instância do atributo na base de dados em questão permite ao usuário outras formas de avaliar a necessidade da seleção do objeto na consulta em questão. A ferramenta VIQUEN disponibiliza os atributos dos objetos em uma janela a parte e não utiliza o recurso de apresentar uma instância da base de dados.

Uma das alterações considerada relevante na nova proposta da interface é o uso da metáfora da sentença para a construção da visão da consulta. Como o

usuário é convidado a selecionar o sujeito da sentença fica transparente para ele o conceito de raiz da consulta para montar a visão desejada na consulta.

A nova interface para a ferramenta VIQUEN procurou explorar todo o potencial expressivo do ambiente VIQUEN para o usuário e propõem a opção pela exigência de um grau mínimo de compreensão de conceitos técnicos da área de banco de dados para a garantia da construção da consulta desejada, uma vez que uma exagerada interferência do sistema em realizar tarefas sem o consentimento do usuário poderia levar a um retorno distante do esperado pelo usuário.

## 7. CONCLUSÃO

Uma linguagem visual para tratamento de base de dados não deve ter somente uma interface apropriada para dar suporte à geração da consulta de forma gráfica, mas deve ser capaz de prover o usuário de um conjunto de estratégias metodológicas para diminuir o esforço cognitivo necessário durante a interação com o sistema. Isto se deve ao fato de que o usuário necessita não só de uma linguagem para a criação da consulta, mas de um meio de comunicação que seja capaz de diminuir a distância semântica gerada por diversas dificuldades que surgem enquanto o usuário está trabalhando em um domínio desconhecido, tais como: descobrir informações implícitas, analisar esquemas muito detalhados, necessidade de armazenar informações pré-requeridas, entre outras. Desta forma, a idéia é oferecer uma ferramenta que apresente um conjunto de primitivas gráficas para auxiliar o usuário no processo de entendimento do esquema da base de dados e que facilite ao máximo a extração das informações relevantes. (Angelaccio et al., 1990 (b))

A literatura revisada mostrou a necessidade de aprofundar a pesquisa de forma a identificar as necessidades adicionais da interface em questão, para se constituir uma interface adequada para a tarefa de construção do sub-esquema da consulta. Isto significa uma interface que demandasse o mínimo de esforço cognitivo para que o usuário entendesse os objetivos do projetista, conforme preconiza a Engenharia Cognitiva descrita por Norman (Norman, 1986) e uma interface que fizesse uso de ícones e signos que chegassem o mais próximo possível da realidade

do usuário, conforme dita a Engenharia Semiótica descrita por De Souza (De Souza, 1993). Alguns tópicos trabalhados na solução proposta são apresentados a seguir.

O uso de signos e ícones para representar a realidade do usuário levou em conta as análises de metáforas feitas por Baranauskas (Oliveira & Baranauskas, 1999) (Baranauskas et al., 1998). Quando se faz uso de signos, ícones e representações gráficas em geral, está sendo utilizada uma Linguagem Visual que deve seguir um formalismo visual para que o usuário consiga identificar as regras de formação desta linguagem mesmo que sendo realizada por meio de gráficos. Esta necessidade fica bem clara nos estudos de Martins (Martins & De Souza, 1998).

Com base nestes estudos foram realizadas análises nas ferramentas visuais acadêmicas de consulta a banco de dados (QBD\*, Pasta-3, SUPER), e com base em métodos acadêmicos de avaliação de interfaces visuais como a LEMD, foi possível avaliar a ferramenta VIQUEN. A partir destas análises foi feita uma proposta de interface visual para a ferramenta VIQUEN, buscando uma metáfora mais apropriada para o processo de consulta a banco de dados visuais, que seria a metáfora da sentença.

O processo de construção da nova interface foi realizado utilizando a metodologia baseada nos conceitos semióticos proposto por De Souza em (De Souza et al., 1999 (a)) e a aplicação dos conceitos de uma linguagem visual sistêmica viabilizou a construção de um resultado consistente através do uso do formalismo da própria LEMD para a construção do modelo funcional da nova interface e do uso da STAG na especificação da estrutura das ações do usuário e respostas do sistema, gerando o mapeamento tarefa-ação. Desta forma foi possível associar dois níveis de definição de representação da interface que se

complementam, permitindo uma melhor definição de modelo de usabilidade do sistema proposto.

Também houve um especial cuidado na elaboração das mensagens de ajuda, procurando uma construção padrão e a utilização de termos conhecidos do usuário do ambiente Windows. As mensagens do designer para o usuário apresentadas na tela foram elaboradas de forma que o usuário consiga se situar nas diversas etapas existentes na aplicação além de ter condições de decidir qual seu próximo passo.

Porém algo que ficou claro neste trabalho, após as análises realizadas nas diversas ferramentas e na linguagem visual proposta para a nova interface, é que no domínio específico de Linguagens Visuais de Consulta a Banco de Dados deve ser exigido um grau mínimo de compreensão de conceitos técnicos para a garantia da construção da consulta desejada, o que reforça os conceitos da ES no sentido da necessidade de uma comunicação designer-usuário eficiente. Outro ponto relevante com relação ao domínio escolhido é que em aplicações que exigem precisão, como é o caso de consulta a base de dados onde a interface permite que o sistema tome iniciativa a partir de ações sobre objetos do diagrama sem a ciência do usuário em relação às suas implicações (o que vem a infringir diretriz gerais do design de interface), não garantem que as ações do sistema sejam efetivamente aquelas desejadas pelo usuário. Nestes sentidos, acreditamos que a opção pela exigência de certo grau de conhecimento de conceitos do modelo de dados utilizado e do domínio da base de dados em questão, aliada a um bom projeto da comunicação do ambiente de interface, incluindo o assistente e os auxílios colocados a disposição do usuário, seja a mais adequada.

## 8. TRABALHOS FUTUROS

Apesar das contribuições já relatadas, alguns conceitos podem ser revistos e algumas características da aplicação podem ser melhoradas, propostas aqui como trabalhos futuros.

Seguindo uma seqüência natural, o próximo trabalho deverá ser o desenvolvimento da interface propriamente dita utilizando o ambiente VIQUEN entre a interface e o SGBD em questão, uma vez que o foco deste trabalho foi o problema de IHC relativo ao processo de seleção do sub-esquema de consulta, e o protótipo desenvolvido focou o primeiro nível de prototipação.

Um outro trabalho a ser realizado consiste no tratamento da aplicação como um todo, integrando as três fases da construção de uma consulta a banco de dados - sub-esquema, predicado e resultado - uma vez que este trabalho deu ênfase somente à etapa de seleção do sub-esquema da consulta. As duas outras fases foram tratadas em trabalhos relacionados. Este esforço encontrará apoio na metodologia utilizada na condução das pesquisas nas três fases, realizada com base no requisito da consistência entre as fases, determinado pela motivação inicial de projeto de um ambiente de interface único.

Finalmente, seria interessante desenvolver uma avaliação da interface proposta utilizando um método de análise diferente da LEMD, que foi a metodologia aplicada para o desenvolvimento da referida interface.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANGELACCIO, Michele. CATARCI, Tiziana. SANTUCCI, Giuseppe. QBD\*: A Fully Visual Query System. In Journal of Visual Languages and Computing, Vol 1, Nº 2, 1990 (a).

ANGELACCIO, Michele. CATARCI, Tiziana. SANTUCCI, Giuseppe. QBD\*: a Graphical Query Language with Recursion. IEEE Transactions on Software Engineering, Vol. 16, Nº 10, 1990 (b).

BARANAUSKAS, M. Cecília C. ROSSLER, Flávia OLIVEIRA, Osvaldo Luiz de . Uma Abordagem Semiótica à Análise de Interfaces: um estudo de caso . Atas do I Workshop sobre Fatores Humanos em Sistemas Computacionais: Compreendendo Usuários, Construindo Interfaces. Maringá/PR. 1998.

BOSCARIOLI, Clodis. DE BASSI, Patricia R. GARCÍA, Laura S. RANTHUM, Geraldo. SUNYE, Marcos S. Visual Databases Languages: A tool evaluation based on the Designer's Message Specification Language. Nimestic 2001 Congrès Scientifique – La relation Homme-Système : Complexe?, Nimes - France, dezembro/2001.

BOSCARIOLI, Clodis. Um Ambiente Semiótico de Interface para Elaboração de Consultas Visuais a Banco de Dados. Proposta de Tese de Mestrado. Departamento de Informática. UFPR. Curitiba-Brasil. 2001.

CARD, S. MORAN, T. NEWELL, A. The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates. Hillsdale, New Jersey. 1983.

CATARCI, Tiziana. COSTABILE, Maria F. CRUZ, Isabel F. IOANNIDIS, Yannis. SHNEIDERMAN, Ben Data Models, Visual Representations, Metaphors: How to

Solve the Puzzle?. IFIP WG2.6 3<sup>rd</sup> Working Conference on Visual Database Systems, March/1995.

CATARZI, Tiziana. COSTABILE, Maria F. LEVIALDI, Stefano. BATINI, Carlo. Visual Query Systems for Databases: a Survey. Dipartimento di Informatica e Sistemistica. Università di Roma "La Sapienza". Roma-Italy, 1997.

DENNENBOUY, Yves. Flexibility of visual languages for data manipulation. IFIP WG2.6 3<sup>rd</sup> Working Conference on Visual Database Systems, March/1995.

DENNENBOUY, Yves. ANDERSSON, Martin. AUDDINO, Annamaria. DUPONT, Yann. FONTANA, Edi. GENTILE, Massimo. SPACCAPIETRA, Stefano. SUPER Visual Interfaces for Object + Relationship Data Models. Journal of Visual Languages and Computing 5, Special Issue on Visual Query Languages, 1995.

DE SOUZA, Clarice Sieckenius. The Semiotic Engineering of User Interface Languages. International Journal of Man-Machine Studies 39, 753-773, 1993.

DE SOUZA, C. S. The Semiotic Engineering of Concreteness and Abstractness: from User Interface Languages to End-User Programming Languages. In Andersen, P. Nadin, M. Nake, F. Informatics and Semiotics. Dagstuhl Seminar Report N° 135, p. 11. Schloss Dagstuhl. Germany, 1996.

DE SOUZA, Clarisse S. LEITE, Jair Cavalcanti. PRATES, Raquel Oliveira. BARBOSA, Simone D.J. Projeto de Interfaces de Usuário – Perspectivas Cognitivas e Semióticas. PUC-Rio. 1999 (a).

DE SOUZA, Clarisse S. PRATES, Raquel O. BARBOSA, Simone D. J. A Method for Evaluating Software Communicability, IHC'99 – II Workshop sobre Fatores Humanos em Sistemas Computacionais, Campinas – SP, Outubro/1999 (b).



DE SOUZA, Clarisse, BARBOSA, Simone D. J., PRATES, Raquel Oliveira. A Semiotic Engineering Approach to User Interface Design. Atas do CHI 2000 – Workshop on Semiotic Approaches to User Design, 2000.

GUEIBER, Ezequiel. Implementação de um Ambiente de Consulta Visual para o Modelo ERC+. Tese de Mestrado. Departamento de Informática. UFPR. Curitiba-Brasil. 2001.

HIX, D. HARTSON, H. Developing User Interfaces: Ensuring Usability Through Products & Process. John Wisley & Sons, 1993.

HUTCHINS, Edwin L. HOLLAN, James D. NORMAN, Donald A. Direct Manipulation Interfaces, In Norman, D. A , Draper, S. W. (eds), User Centered System Design. Hillsdale/NJ. Lawrence Erlbaum Associates, capítulo 5, 1986.

KUNTZ, Michel. MELCHERT, Rainer. Pasta-3's Graphical Query Language: Direct Manipulation, Cooperative Queries, Full Expressive Power. Proceedings of the Fifteenth International Conference on Very Large Data Bases. Amsterdam, 1989.

LEITE, Jair Cavalcanti. Modelos e Formalismos para a Engenharia Semiótica de Interface de Usuário. Tese de doutorado. PUC-Rio, 1998.

LEITE, Jair C . DE SOUZA, Clarisse S. Uma Linguagem de Especificação para a Engenharia Semiótica de Interfaces de Usuários. Atas do II Workshop sobre Fatores Humanos em Sistemas Computacionais–IHC'1999, Campinas – SP, 1999.

MARTINS, Isa Haro. Um Instrumento de Análise Semiótica para Linguagens Visuais de Interfaces. Tese de Doutorado, Departamento de Informática, PUC-RJ, Abril/1998.

MARTINS, Isa Haro. DE SOUZA, Clarisse S. Uma Abordagem Semiótica na Utilização dos Recursos Visuais em Linguagens de Interface . Atas do I Workshop

sobre Fatores Humanos em Sistemas Computacionais: Compreendendo Usuários, Construindo Interfaces. Maringá/PR. 1998.

NADIN, Mihai, One Cannot Not Interact, CHI'2000 Workshop on Semiotic Approaches to User Interface Design, Abril-2000.

NORMAN, Donald A. Cognitive Engeneering , In Norman, D. A , Draper, S. W. (eds), User Centered System Design. Hillsdale/NJ. Lawrence Erlbaum Associates, capítulo 3, 1986.

OLIVEIRA, Osvaldo Luiz de. BARANAUSKAS, M. Cecília Calani. A Semiótica e o Design de Software, Relatório Técnico IC-98-09, Abril/1998 (a).

OLIVEIRA, Osvaldo Luiz de. BARANAUSKAS, M. Cecília Calani. Análise Glossemática da Estrutura das Linguagens de Interface Humano-Computador, Atas do I Workshop sobre Fatores Humanos em Sistemas Computacionais – IHC'98, Maringá, PR, Outubro/1998 (b).

OLIVEIRA, Osvaldo Luiz de. BARANAUSKAS, M. Cecília C. Interface Entendida Como um Espaço de Comunicação. Atas do II Workshop sobre Fatores Humanos em Sistemas Computacionais: Rompendo Barreiras entre Pessoas e Computadores. Campinas/SP. 1999.

PAYNE, Stephen J. GREEN, T. R. G. Task-Action Grammars: A Model of the Mental Representation of Task Languages, Human-Computer Interaction, volume 2, Xerox Palo Alto Research Center, Palo Alto- CA, 1986.

PIMENTEL, Andrey R. DIRENE, Alexandre I. Medidas Cognitivas no Ensino de Programação de Computadores com Sistemas Tutores Inteligentes. Anais do Simpósio Brasileiro de Informática na Educação - SBIE, 1998.

PIMENTEL, Andrey R. DIRENE, Alexandre I. Medidas Cognitivas para o Ensino de Conceitos Visuais com Sistemas Tutores Inteligentes. Anais do Simpósio Brasileiro de Informática na Educação - SBIE, 1997.

PRADO, A. B. BARANAUSKAS, M. C. C. Projeto Granel – Investigando possibilidades da abordagem semiótica em design de interfaces. Atas do II Workshop sobre Fatores Humanos em Sistemas Computacionais–IHC'1999, Campinas – SP, 1999.

PRADO, Alysson Bolognesi. BARANAUSKAS, Maria Cecília Calani. Avaliando a Meta-comunicação Designer-Usuário de Interface. Atas do III Workshop sobre Muitas Faces em Interfaces – IHC'2000, Unisinos – RS, 2000.

PREECE, J. ROGERS, Y. SHARP, H. BENYON, D. HOLLAND, S. CARCY, T. Human-Computer Interaction. Addison-Wesley, 1994.

RANTHUM, Geraldo. Um Ambiente de Interface para Visualização do Resultado das Consultas a um Banco de Dados. Proposta de Tese de Mestrado. Departamento de Informática. UFPR. Curitiba-Brasil. 2001.

RODACKI, Andréa. Aplicação de Estratégias de Integração de Banco de Dados: um Estudo de Caso. Tese de Mestrado. Departamento de Informática. UFPR. Curitiba-Brasil. 2000.

SPACCAPIETRA, Stefano. PARENT, Christine. ERC+: An Object Based Entity Relationship Approach. In *Conceptual Modelling Database and Case: An Integrated View of Information System Development*. Peri Loucopoulos and Roberto Zicari Eds, John Wiley, 1992.

SPACCAPIETRA, Stefano. PARENT, Christine. SUNYE Marcos. YETONGNON, Kokou. LEVA, Antonio Di. An Object + Relationship Paradigm for Database Applications. Readings in Object-Oriented Systems, D. Rine (Ed.), IEEE Press, 1995.

ZLOOF, M. M. Query-by-Example: A Data Base Language. In IBM Systems Journal, vol 16, n° 4, 1977.

## APÊNDICE 1 – A LEMD DA INTERFACE

### 1. Regras de mapeamento semântico

Elemento da LEMD	<i>Widget</i> associado	Significado
<i>View</i>	Painéis ou message-boxes	Resultado apresentado pelo sistema referente ao acionamento de uma função da aplicação realizada pelo usuário
<i>Text</i>	Labels ou message-boxes	Mensagem direta do designer para o usuário
<i>Activate</i>	Botões de pressão	Acionamento de uma função da aplicação
<i>Select information-of</i>	Combo-boxes ou painéis	Escolha de uma opção de uma lista ou no diagrama apresentado
<i>Enter</i>	Caixa de texto	Digitação de mensagem a ser enviada ao sistema
<i>Join</i>	Painéis ou boxes	Agrupamento
<i>Repeat</i>	Painéis ou boxes	A tarefa apresentada pode ser executada mais de uma vez
<i>Sequence</i>	Painéis ou boxes	Seqüência de execução das tarefas apresentadas não é obrigatória
<i>Combine</i>	Painéis ou boxes	Combinação de execução das tarefas apresentadas é obrigatória e fixa
<i>Select</i>	Botões	Tarefa do usuário de escolher uma ação a ser executada pelo sistema.
<i>Command-message</i>	Formulário	Agrupamento de tarefas do usuário e ações do sistema
<i>Task-message</i>	Janela	Agrupamento auxiliar de tarefa do usuário e ações do sistema.
<i>Control-message</i>	Janela	Agrupamento de mensagem direta do designer para o usuário, tarefa do usuário e ações do sistema.

### 2. LEMD

Command-Message Bem-Vindo ao Sistema for application function Recuperar Esquema

```

Combine {
  Join {
    Text "Digitar os dados de identificação do usuário."
      View "Usuário"
      Enter information-of Nome do Usuário
      View "Senha"
      Enter information-of Senha do Usuário
    Text "Selecionar o esquema desejado para a consulta"
      View "Esquema"
      Select information-of <Drive onde o arquivo da base de dados se encontra>
      Select information-of <Nome dos arquivos de base de dados e sub-esquemas gravados>
  }
  Join {
    Select {
      Activate Start application-function Recuperar-esquema
      Activate Start application-function Ajuda
    }
  }
} //fim Esquema

```

Command-Message Construção do Sub-Esquema for application-function Construção do Sub-Esquema da Consulta

```

Combine {
  Command-Message Seleção da Raiz da Consulta for application-function Selecionar Raiz

```

```

Combine {
  Join {
    Text "Selecionar o objeto que será a raiz da consulta, funcionando como o sujeito da Sentença a ser criada."
    View <esquema relacional selecionado trazido para o modelo ERC+ - esquema original>
    Select information-of <esquema original>
  }
  Join {
    Select {
      Activate Waive application-function Cancelar
      Activate Suspend application-function Selecionar raiz
      Activate Start application-function Selecionar raiz
      Activate Start application-function Ajuda
    }
  }
}
} //fim seleção-raiz

```

Command-Message Seleção dos Objetos for application function Selecionar dos Objetos

```

Repeat {
  Combine{
    Join{
      Text "Esquema tema da raiz da consulta, que vai funcionar como o sujeito da sentença construída"

```

Text *"Selecionar os objetos que irão compor a consulta funcionando como o predicado da sentença construída"*

View <esquema original em uma estrutura de sentença deslocando a raiz da consulta como o sujeito da sentença – esquema original em forma de sentença>

Select information-of < esquema original em forma de sentença>

}

Task-Message Seleção dos Objetos – janela auxiliar

Sequence {

Join {

Text *"Os objetos selecionados farão parte do novo sub-esquema"*

Text *"Selecionar os objetos desejados no esquema original."*

Select {

Activate Start application function Selecionar Todos

Activate Start application-function Retirar

}

View <nome dos objetos selecionados>

}

Join {

Select {

Activate Waive application-function Cancelar

Activate Start application function Confirmar

Activate Start application-function Ajuda

}



```

    }
} //fim seleciona-objetos-janela-auxiliar

Join {
  Select {
    Activate Waive application-function Cancelar
    Activate Suspend application function Selecionar objetos
    Activate Start application-function Selecionar objetos
    Activate Start application function Ajuda
  }
}
} //combine
} // repeat
Control-Message
Join {
  Text "Deseja selecionar mais algum objeto?"
  Select {
    Activate Start application function Selecionar objetos
    Activate Stop application-function Selecionar objetos
  }
}
} //fim seleciona objetos

```

Command-Message Elimina Ciclos for application-function Eliminar Ciclos

```

Repeat {
  Combine {
    Join {
      Text "Sub-esquema selecionado, funcionando como a sentença construída"
      Text "Eliminar as ligações que estão formando um círculo na sentença construída."
      View <visão do sub-esquema selecionado pelo usuário em estrutura hierárquica, utilizando o
      objeto raiz selecionado como sujeito da sentença. Onde os caminhos necessários para manter a
      integridade da consulta foram selecionados automaticamente pelo sistema sem intervenção do
      usuário>
      Select information-of <visão do sub-esquema em forma de sentença>
    }
    Join {
      Select {
        Activate Waive application-function Cancelar
        Activate Suspend application function Eliminar ciclo
        Activate Start application-function Eliminar ciclo
        Activate Start application-function Adição de Novos Elementos
        Activate Start application function Ajuda
      }
    }
  } // combine
} //repeat
Control-message
Join {

```

Text "Deseja eliminar *mais algum ciclo?*"

Select {

    Activate Start application function Eliminar ciclo

    Activate Stop application-function Eliminar ciclo

    }

}

// fim elimina-ciclos

Command-Message Adição de novos elementos for application-function Adição de Novos Elementos

Repeat {

    Sequence{

        Join {

            Text "Sub-esquema selecionado sem ciclos, funcionando como a sentença construída"

            Text "*Selecionar novos objetos ou novo sub-esquema, se desejado.*"

            View <visão do sub-esquema em forma de sentença sem o ciclo selecionado>

        }

        Join {

            Select {

                Activate Waive application-function Cancelar

                Activate Suspend application function Adição de novos elementos

                Activate Start application-function Seleção de novos objetos

                Activate Start application-function Seleção de novo sub-esquema

                Activate Start application-function Seleção de Atributos

                Activate Start application function Ajuda

```

    }
  }
} //sequence
} // fim adiciona elementos

```

Command-Message Seleção de Novo Sub-Esquema for application-function Novo Sub-Esquema

```

Combine {
  Join {
    Text "Selecionar um novo sub-esquema já gravado para adicionar à nova consulta a ser
    construída."
    Text "Nome do Sub-esquema"
    View <relação de nome de arquivos de sub-esquemas gravados referentes à mesma base
    de dados do esquema original>
    Select information-of <arquivos de sub-esquemas já gravados>
  }
  Join {
    Select {
      Activate Waive application-function Cancelar
      Activate Stop application function Seleção de novo subesquema
      Activate Start application-function Unificação de novos elementos
      Activate Start application-function Adição de novos elementos
      Activate Start application function Ajuda
    }
  }
}

```

```
}// fim novo-subesquema
```

Command-Message Seleção de Novos Objetos for application function Selecionar Objetos

```
Repeat {
  Sequence{
    Join{
      Text "Esquema tema da raiz da consulta, que vai funcionar como o sujeito da sentença
construída"
      Text "Selecionar os novos objetos além dos já selecionados, que formarão o predicado da
sentença construída."
      View <esquema original em uma estrutura de sentença deslocando a raiz da consulta
como o sujeito da sentença e os objetos já selecionados marcados – esquema original em
forma de sentença>
      Select information-of < esquema original em forma de sentença >
    }
  }
}
```

Task-Message Seleção de Novos Objetos – janela auxiliar

```
Sequence {
  Join {
    Text "Os objetos selecionados farão parte do novo sub-esquema"
    Text "Selecionar os objetos desejados no esquema original."
    Select {
      Activate Start application function Selecionar Todos
      Activate Start application-function Retirar
    }
  }
}
```

```

    }
    View <nome dos objetos selecionados>
  }
  Join {
    Select {
      Activate Waive application-function Cancelar
      Activate Start application function Confirmar
      Activate Start application-function Ajuda
    }
  }
} //fim seleciona-novos-objetos-janela-auxiliar

Join {
  Select {
    Activate Waive application-function Cancelar
    Activate Suspend application function Selecionar objetos
    Activate Start application-function Selecionar objetos
    Activate Start application function Ajuda
  }
}
} //sequence
} //repeat

Control-Message

```

```

Join {
  Text "Deseja selecionar mais algum objeto?"
  Select {
    Activate Start application function Selecionar objetos
    Activate Stop application-function Selecionar objetos
  }
}
//fim seleciona objetos

```

Command-Message Unificação de Novos Elementos for application-function Unificar Sub-Esquema

```

Repeat {
  Combine {
    Join {
      Text "Sub-esquema selecionado com os novos objetos e/ou sub-esquema"
      Text "Unificar os objetos e/ou sub-esquemas para conectar todos em um só sub-esquema. "
      Select information-of <visão do sub-esquema com os novos objetos e/ou sub-esquemas
      selecionados>
    }
  }
  Join {
    Select {
      Activate Waive application-function Cancelar
      Activate Stop application function Unificar sub-esquema
      Activate Start application-function Unificar sub-esquema
      Activate Start application-function Seleção de Atributos
    }
  }
}

```

```

        Activate Start application function Ajuda
    }
}
} //combine
} //repeat

Control-message
  Join {
    Text "Deseja unificar mais algum sub-esquema?"
    Select {
      Activate Start application function Unificar sub-esquema
      Activate Stop application-function Unificar sub-esquema
    }
  }
  // fim unificação-subesquema

Command-message Seleção de Atributos for application-function Selecionar de Atributos
  Repeat{
    Combine{
      Join {
        Text "Sub-esquema selecionado funcionando como a sentença construída."
        Text "Selecionar o objeto para a seleção de seus atributos. Os atributos selecionados
        farão parte da consulta."
        View <visão do sub-esquema adicionado>
      }
    }
  }

```



```

    Select information-of <visão do sub-esquema adicionado>
  }

```

Task-Message Seleção dos Atributos – janela auxiliar

```

Sequence {
  Join {
    Text "Selecionar os atributos desejados no sub-esquema."
    Select {
      Activate Start application function Selecionar Todos
      Activate Start application-function Retirar
    }
    View <nome dos atributos selecionados>
  }
  Join {
    Select {
      Activate Waive application-function Cancelar
      Activate Start application function Confirmar
      Activate Start application-function Ajuda
    }
  }
}
} //fim seleciona-atributos-janela-auxiliar

```

```

Join {
  Select {

```

```
        Activate Waive application-function Cancelar
        Activate Stop application function Selecionar atributos
        Activate Start application-function Selecionar atributos
        Activate Start application function Ajuda
    }
}
} //combine
} // fim seleciona-atributos

Command-message Ajuda for Application-function Ajuda
Join {
    Text <texto de ajuda referente ao item contendo: como realizar a operação>
    Activate Discard Application-function Ajuda
}
// fim ajuda

} //fim construção do sub-esquema
```

## APÊNDICE 2 – A STAG DA INTERFACE

### Declaração de tipos e instâncias

#### 1.1. Tipos de Dados do Sistema (entidades, atributos e relacionamentos)

primitiva

primitiva-interação

primitiva-composta

plano-interação

#### 1.2. Instâncias de Dados do Sistema (entidades, atributos e relacionamentos)

primitiva: entidade, relacionamento, cardinalidade (opcional monovalorado 0:1), cardinalidade (monovalorado e mandatário 1:1), cardinalidade (opcional e multivalorado 0:N), cardinalidade (mandatária e multivalorado 1:N), atributo, atributo\_complexo, nome de arquivo, drive de periférico

primitiva-interação: diagrama\_subesquema, raiz\_subesquema, objeto\_subesquema\_usuario, objeto\_subesquema\_sistema, diagrama\_novo\_subesquema

primitiva-composta: esquema\_original, subesquema

plano-interação: plano

#### 1.3. Tipos de Operações

operações-edição

#### 1.4. Instâncias de Operações

operações-edição: digitar, eliminar, unificar, copiar

#### 1.5. Tipos de Variáveis de Estado das Entidades

estado-seleção

#### 1.6. Instâncias de Variáveis de Estado das Entidades

estado-seleção: ativo, inativo

#### 1.7. Relações de Sobreposição

primitiva ← estado-seleção

#### 1.8. Tipos de Expressão do Sistema

forma

forma-composta

diagrama

cor

texto

#### 1.9. Instâncias de Expressão do Sistema

forma: retângulo, losango, linhas cheias e/ou tracejadas, palavra

forma-composta: janela-rolagem, janela, botão

diagrama: retângulo, losango, linhas cheias e/ou tracejadas, palavra

cor: preto, laranja, azul-claro, azul-escuro

texto: menu-hierárquico, rótulo

#### 1.10. Tipos do Conteúdo do Usuário

intenção

#### 1.11. Instâncias do Conteúdo do Usuário

intenção: digitar, posicionar, selecionar, pressionar

#### 1.12. Tipos da Expressão do Usuário

sinal

#### 1.13. Instâncias da Expressão do Usuário

sinal: clique-botão-esquerdo, pressão-tecla, parar-com-mouse

### 1. Mapeamento entre elementos do conteúdo e da expressão

#### 2.1. Mapeamento de Tipos do Sistema

primitiva  $\Leftrightarrow$  forma

primitiva-interação  $\Leftrightarrow$  cor

estado-seleção  $\Leftrightarrow$  cor  $\Leftrightarrow$  forma

primitiva-composta  $\Leftrightarrow$  diagrama

plano-interação  $\Leftrightarrow$  diagrama  $\Leftrightarrow$  cor

#### 2.2. Mapeamento de Instâncias do Sistema

primitiva (entidade)  $\Leftrightarrow$  forma (retângulo)

primitiva (relacionamento)  $\Leftrightarrow$  forma (losango)

primitiva (cardinalidade (monovalorado e mandatório 1:1))  $\Leftrightarrow$  forma (linha simples contínua)

primitiva (cardinalidade (opcional e monovalorado 0:1))  $\Leftrightarrow$  forma (linha simples tracejada)

primitiva (cardinalidade (opcional e multivalorado 0:N))  $\Leftrightarrow$  forma (linha dupla tracejada)

primitiva (cardinalidade (mandatório e multivalorado 1:N))  $\Leftrightarrow$  forma (linha tracejada e linha contínua)

primitiva (atributo)  $\Leftrightarrow$  forma (palavra)

primitiva (atributo\_complexo)  $\Leftrightarrow$  forma (palavra)

primitiva (nome de arquivo)  $\Leftrightarrow$  forma (palavra)

primitiva (drive de periférico)  $\Leftrightarrow$  forma (palavra)

diagrama {@}  $\Leftrightarrow$  forma (retângulo, losango, linhas cheias e/ou pontilhadas, palavra)

primitiva-interação (raiz\_subesquema)  $\Leftrightarrow$  cor (laranja)

primitiva-interação (diagrama\_subesquema)  $\Leftrightarrow$  cor (preto)

primitiva-interação (objeto\_subesquema\_usuario)  $\Leftrightarrow$  cor (azul-claro)

primitiva-interação (objeto\_subesquema\_sistema)  $\Leftrightarrow$  cor (azul-escuro)  
 primitiva-interação (diagrama\_novo\_subesquema)  $\Leftrightarrow$  cor (azul-escuro)  
 estado-seleção (ativo)  $\Leftrightarrow$  cor (laranja)  $\langle \rangle$  forma (retângulo/losango)  
 estado-seleção (ativo)  $\Leftrightarrow$  cor (azul-claro)  $\langle \rangle$  forma (retângulo/losango)  
 estado-seleção (ativo)  $\Leftrightarrow$  cor (azul-claro)  $\langle \rangle$  forma (linhas cheias e/ou pontilhadas)  
 estado-seleção (ativo)  $\Leftrightarrow$  cor (azul-claro)  $\langle \rangle$  forma (palavra)  
 estado-seleção (inativo)  $\Leftrightarrow$  nulo

### 2.3. Mapeamento de Tipos do Usuário

intenção  $\Leftrightarrow$  sinal

### 2.4. Mapeamento de Instâncias do Usuário

intenção (digitar)  $\Leftrightarrow$  sinal (pressão-tecla)

intenção (posicionar)  $\Leftrightarrow$  sinal (parar-com-mouse)

intenção (pressionar)  $\Leftrightarrow$  sinal (parar-com-mouse + clique-botão-esquerdo)

intenção (selecionar)  $\Leftrightarrow$  sinal (parar-com-mouse + clique-botão-esquerdo)

## 2. Dicionário de tarefas simples

Digitar-objeto [operação = digitar, objeto]

Selecionar-objeto [operação = selecionar, objeto]

Eliminar-objeto [operação = eliminar, objeto]

## 3. Regras gramaticais

Digitar-objeto [operação = digitar, objeto]  $\rightarrow$  primitiva (texto) + intenção (digitar) + operação-edição (digitar)

Selecionar-nome [operação = selecionar, objeto]  $\rightarrow$  primitiva (palavra) + intenção (selecionar) + estado-seleção (ativo)

Selecionar-objeto [operação = selecionar, objeto]  $\rightarrow$  primitiva (forma) + intenção (selecionar) + estado-seleção (ativo)

Pressionar-botão [operação = selecionar, objeto]  $\rightarrow$  primitiva (forma-composta) + intenção (pressionar) + estado-seleção (ativo)

Selecionar-esquema [operação = selecionar, objeto]  $\rightarrow$  selecionar-nome [operação = selecionar, objeto = nome do arquivo] + operação-edição (copiar)

Selecionar-raiz [operação = selecionar, objeto] → selecionar-objeto  
[operação=selecionar, objeto=entidade/relacionamento] + operação-edição  
(copiar)

Retirar-seleção-objeto [operação = eliminar, objeto] → selecionar-objeto  
[operação = selecionar, objeto = entidade/relacionamento] + operação-edição  
(eliminar)

Eliminar-ciclo [operação = eliminar, objeto] → selecionar-objeto [operação =  
selecionar, objeto = linhas cheias e/ou tracejadas] + operação-edição (eliminar)

Adicionar-objeto [operação = selecionar, objeto] → selecionar-objeto  
[operação=selecionar, objeto = entidade/relacionamento] + operação-edição  
(copiar)

Adicionar-sub-esquema [operação = selecionar, objeto] → selecionar-nome  
[operação =selecionar, objeto = nome do arquivo] + operação-edição (copiar)

Unificar-subesquema [operação = selecionar, objeto] → selecionar-objeto  
[operação=selecionar, objeto = entidade] + operação-edição (unificar)

Selecionar-atributo [operação = selecionar, objeto] → selecionar-nome  
[operação=selecionar, objeto=palavra] + operação-edição (copiar)

Retirar-seleção-atributo [operação = eliminar, objeto] → selecionar-atributo  
[operação = selecionar, objeto = palavra] + operação-edição (eliminar)

## APÊNDICE 3 – O AUXÍLIO *ONLINE* DA INTERFACE

Funcionamento da interface – a interface esta baseada no ambiente Windows e oferece os recursos do mesmo com relação ao uso do mouse, seleção de objetos na tela, identificação de tarefa disponível para execução, *hint* nos ícones.

Descrição geral da interface – a interface é composta por janelas que são abertas de maneira sucessiva a medida em que o usuário avança nas etapas e sub-etapas de construção da consulta.

Conceitos de objetos e funções da interface

Janela1: Barra de título –

Bem Vindo ao Sistema

Barra de menu –

Arquivo

Abrir

Fechar

Salvar

Salvar como

Sair

Sub-Esquema

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

Predicado

Resposta

Ajuda

Sobre o Windows

Sobre o Viquen

Sobre Banco de Dados

Botões de funções e botões de *status* -

Identificação - Usuário

Localização na tela - Primeiro painel parte superior esquerda

Procedimento de seleção e ativação - Digitar o nome do usuário cadastrado no sistema

Identificação - Senha

Localização na tela - Primeiro painel parte superior direita

Procedimento de seleção e ativação - Digitar a senha do usuário cadastrada no sistema

Identificação - Esquema

Localização na tela - Primeiro painel parte inferior primeiro *box*

Procedimento de seleção e ativação - Posicionar o cursor no ícone de seta, pressionar e soltar quando aparecer na lista apresentada o nome do *drive* do sistema onde se encontra a base de dados do domínio da consulta desejada. Posicionar o cursor no nome do *drive* desejado e pressionar o botão esquerdo do mouse e soltar.

Identificação - Esquema

Localização na tela - Primeiro painel parte inferior segundo *box*

Procedimento de seleção e ativação - Posicionar o cursor no ícone de seta, pressionar e soltar quando aparecer na lista apresentada o nome do arquivo onde se encontra a base de dados do domínio da consulta desejada. Posicionar o cursor no nome do arquivo desejado e pressionar o botão esquerdo do mouse e soltar.

Identificação - Recuperar Esquema

Localização na tela - Segundo painel lado esquerdo

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Ajuda

Localização na tela - Segundo painel lado direito

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.



Janela2: Barra de título –

Construção do Sub-Esquema Seleção da Raiz da Consulta

Barra de menu –

Arquivo

Abrir

Fechar

Salvar

Salvar como

Sair

Sub-Esquema

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

Predicado

Resposta

Ajuda

Sobre o Windows

Sobre o Viquen

Sobre Banco de Dados

Botões de funções e botões de *status* -

Identificação - Figura do esquema original do domínio da base de dados

Localização na tela - Primeiro painel parte superior

Posicionar o cursor no objeto desejado da figura, pressionar o botão esquerdo do mouse e soltar.

Identificação - Cancelar

Localização na tela - Segundo painel primeiro botão

Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Etapa Anterior

Localização na tela - Segundo painel segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Selecciona Raiz**

Localização na tela - Segundo painel terceiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Ajuda**

Localização na tela - Segundo painel quarto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Janela3: Barra de título –**

**Construção do Sub-Esquema Seleção dos Objetos**

**Barra de menu –**

**Arquivo**

Abrir

Fechar

Salvar

Salvar como

Sair

**Sub-Esquema**

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

**Predicado**

**Resposta**

**Ajuda**

Sobre o Windows

Sobre o Víquen

Sobre Banco de Dados

**Botões de funções e botões de *status* -**

**Identificação** - Figura do esquema tema da raiz da consulta que funciona como o sujeito da sentença construída

**Localização na tela** - Primeiro painel parte superior

**Procedimento de seleção e ativação** - Posicionar o cursor no objeto desejado da figura, pressionar o botão esquerdo do mouse e soltar.

**Identificação** - Cancelar

**Localização na tela** - Segundo painel primeiro botão

**Procedimento de seleção e ativação** - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação** - Etapa Anterior

**Localização na tela** - Segundo painel segundo botão

**Procedimento de seleção e ativação** - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação** - Seleciona Objeto

**Localização na tela** - Segundo painel terceiro botão

**Procedimento de seleção e ativação** - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação** - Próximo Passo

**Localização na tela** - Segundo painel quarto botão

**Procedimento de seleção e ativação** - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação** - Ajuda

**Localização na tela** - Segundo painel quinto botão

**Procedimento de seleção e ativação** - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Sub-janela 3.1:

**Barra de título –**

**Construção do Sub-Esquema Seleção dos  
Objetos**

**Botões de funções e botões de *status* -**

**Identificação - Selecionar Todos**

**Localização na tela - Primeiro painel primeiro  
botão**

**Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.**

**Identificação - Retirar**

**Localização na tela - Primeiro painel segundo  
botão**

**Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.**

**Identificação - Lista de itens selecionados**

**Localização na tela - Primeiro painel lado  
direito**

**Identificação - Cancelar**

**Localização na tela - Segundo painel primeiro  
botão**

**Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.**

**Identificação - Confirmar**

**Localização na tela - Segundo painel segundo  
botão**

**Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.**

**Identificação - Ajuda**

**Localização na tela - Segundo painel terceiro  
botão**

Procedimento de seleção e ativação -  
 Posicionar o cursor no botão, pressionar o  
 botão esquerdo do mouse e soltar.

**Sub-janela 3.2:**

Barra de título –

Pr-Viquen

Botões de funções e botões de *status* -

Identificação - Yes

Localização na tela - Primeiro botão

Procedimento de seleção e ativação -  
 Posicionar o cursor no botão, pressionar o  
 botão esquerdo do mouse e soltar.

Identificação - No

Localização na tela - Segundo botão

Procedimento de seleção e ativação -  
 Posicionar o cursor no botão, pressionar o  
 botão esquerdo do mouse e soltar.

**Janela4: Barra de título –**

Construção do Sub-Esquema Elimina Ciclos

Barra de menu –

Arquivo

Abrir

Fechar

Salvar

Salvar como

Sair

Sub-Esquema

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

Predicado

Resposta

Ajuda

Sobre o Windows

## Sobre o Viquen

### Sobre Banco de Dados

#### Botões de funções e botões de status -

Identificação - Figura do sub-esquema selecionado funcionando como a sentença construída

Localização na tela - Primeiro painel parte superior

Procedimento de seleção e ativação - Posicionar o cursor na ligação desejada da figura, pressionar o botão esquerdo do mouse e soltar.

Identificação - Cancelar

Localização na tela - Segundo painel primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Etapa Anterior

Localização na tela - Segundo painel segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Elimina Ciclo

Localização na tela - Segundo painel terceiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Próximo Passo

Localização na tela - Segundo painel quarto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Ajuda

Localização na tela - Segundo painel quinto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Janela5: Barra de título –

Construção do Sub-Esquema Adição de Novos Elementos

Barra de menu –

Arquivo

Abrir

Fechar

Salvar

Salvar como

Sair

Sub-Esquema

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

Predicado

Resposta

Ajuda

Sobre o Windows

Sobre o Viquen

Sobre Banco de Dados

Botões de funções e botões de *status* -

Identificação - Figura do sub-esquema selecionado sem ciclos funcionando como a sentença construída

Localização na tela - Primeiro painel parte superior

Identificação - Cancelar

Localização na tela - Segundo painel primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Etapa Anterior

Localização na tela - Segundo painel segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Objetos**

Localização na tela - Segundo painel terceiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Sub-Esquema**

Localização na tela - Segundo painel quarto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Próximo Passo**

Localização na tela - Segundo painel quinto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Ajuda**

Localização na tela - Segundo painel sexto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Sub-janela 5.1:**

Barra de título –

Pr-Viquen

Botões de funções e botões de *status* -

Identificação - Yes

Localização na tela - Primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - No

Localização na tela - Segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.



Janela6: Barra de título –

Construção do Sub-Esquema Seleção de Novo Sub-Esquema

Barra de menu –

Arquivo

Abrir

Fechar

Salvar

Salvar como

Sair

Sub-Esquema

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

Predicado

Resposta

Ajuda

Sobre o Windows

Sobre o Viquen

Sobre Banco de Dados

Botões de funções e botões de *status* -

Identificação - Nome do Sub-Esquema

Localização na tela - Primeiro painel

Procedimento de seleção e ativação - Posicionar o cursor no ícone de seta, pressionar e soltar quando aparecer na lista apresentada o nome do arquivo onde se encontra o sub-esquema desejado. Posicionar o cursor no nome do arquivo desejado e pressionar o botão esquerdo do mouse e soltar.

Identificação - Cancelar

Localização na tela - Segundo painel primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Etapa Anterior**

Localização na tela - Segundo painel segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Novo Sub-Esquema**

Localização na tela - Segundo painel terceiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Próximo Passo**

Localização na tela - Segundo painel quarto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Ajuda**

Localização na tela - Segundo painel quinto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Janela7: Barra de título –**

Construção do Sub-Esquema Seleção de Novos dos  
Objetos

**Barra de menu –****Arquivo**

Abrir

Fechar

Salvar

Salvar como

Sair

**Sub-Esquema**

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

Predicado

Resposta

Ajuda

Sobre o Windows

Sobre o Viquen

Sobre Banco de Dados

Botões de funções e botões de *status* -

Identificação - Figura do esquema tema da raiz da consulta que funciona como o sujeito da sentença construída com os objetos já selecionados marcados

Localização na tela - Primeiro painel parte superior

Procedimento de seleção e ativação - Posicionar o cursor no objeto desejado da figura, pressionar o botão esquerdo do mouse e soltar.

Identificação - Cancelar

Localização na tela - Segundo painel primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Etapa Anterior

Localização na tela - Segundo painel segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Seleciona Objeto

Localização na tela - Segundo painel terceiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Próximo Passo

Localização na tela - Segundo painel quarto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Ajuda

Localização na tela - Segundo painel quinto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Sub-janela 7.1:

Barra de título –

Construção do Sub-Esquema Seleção de Novos Objetos

Botões de funções e botões de *status* -

Identificação - Selecionar Todos

Localização na tela - Primeiro painel primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Retirar

Localização na tela - Primeiro painel segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Lista de itens selecionados

Localização na tela - Primeiro painel lado direito

Identificação - Cancelar

Localização na tela - Segundo painel primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Confirmar

Localização na tela - Segundo painel segundo botão

Procedimento de seleção e ativação -  
 Posicionar o cursor no botão, pressionar o  
 botão esquerdo do mouse e soltar.

Identificação - Ajuda

Localização na tela - Segundo painel terceiro  
 botão

Procedimento de seleção e ativação -  
 Posicionar o cursor no botão, pressionar o  
 botão esquerdo do mouse e soltar.

Janela8: Barra de título –

Construção do Sub-Eschema Unificação dos Objetos  
 e/ou Sub-Esquemas

Barra de menu –

Arquivo

Abrir

Fechar

Salvar

Salvar como

Sair

Sub-Eschema

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

Predicado

Resposta

Ajuda

Sobre o Windows

Sobre o Viquen

Sobre Banco de Dados

Botões de funções e botões de *status* -

Identificação - Figura do sub-esquema selecionado  
 com os novos objetos e/ou sub-esquema

Localização na tela - Primeiro painel parte superior

Procedimento de seleção e ativação - Posicionar o cursor no objeto desejado da figura, pressionar o botão esquerdo do mouse e soltar.

Identificação - Cancelar

Localização na tela - Segundo painel primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Etapa Anterior

Localização na tela - Segundo painel segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Unificar Sub-Esquema

Localização na tela - Segundo painel terceiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Próximo Passo

Localização na tela - Segundo painel quarto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Identificação - Ajuda

Localização na tela - Segundo painel quinto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

Sub-janela 8.1:

Barra de título –

Pr-Viquen

Botões de funções e botões de *status* -

Identificação - Yes

Primeiro botão

Procedimento de seleção e ativação -  
 Posicionar o cursor no botão, pressionar o  
 botão esquerdo do mouse e soltar.

Identificação - No

Localização na tela - Segundo botão

Procedimento de seleção e ativação -  
 Posicionar o cursor no botão, pressionar o  
 botão esquerdo do mouse e soltar.

Janela9: Barra de título –

Construção do Sub-Esquema Seleção de Atributos

Barra de menu –

Arquivo

Abrir

Fechar

Salvar

Salvar como

Sair

Sub-Esquema

Selecionar Raiz da Consulta

Selecionar Objetos

Eliminar Ciclos

Adicionar Novos Elementos

Unificar Novos Elementos

Selecionar Atributos

Predicado

Resposta

Ajuda

Sobre o Windows

Sobre o Viquen

Sobre Banco de Dados

Botões de funções e botões de *status* -

Identificação - Figura do sub-esquema selecionado  
 com os novos objetos e/ou sub-esquema

Localização na tela - Primeiro painel parte superior

Procedimento de seleção e ativação - Posicionar o  
 cursor no objeto desejado da figura, pressionar o botão  
 esquerdo do mouse e soltar.

**Identificação - Cancelar**

Localização na tela - Segundo painel primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Etapa Anterior**

Localização na tela - Segundo painel segundo botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Seleciona Atributos**

Localização na tela - Segundo painel terceiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Próximo Passo**

Localização na tela - Segundo painel quarto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Identificação - Ajuda**

Localização na tela - Segundo painel quinto botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.

**Sub-janela 9.1:**

Barra de título –

Construção do Sub-Esquema Seleção dos Atributos

Botões de funções e botões de *status* -

Identificação - Selecionar Todos

Localização na tela - Primeiro painel primeiro botão



Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.

Identificação - Retirar

Localização na tela - Primeiro painel segundo  
botão

Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.

Identificação - Lista de itens selecionados

Localização na tela - Primeiro painel lado  
direito

Identificação - Cancelar

Localização na tela - Segundo painel primeiro  
botão

Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.

Identificação - Confirmar

Localização na tela - Segundo painel segundo  
botão

Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.

Identificação - Ajuda

Localização na tela - Segundo painel terceiro  
botão

Procedimento de seleção e ativação -  
Posicionar o cursor no botão, pressionar o  
botão esquerdo do mouse e soltar.

Janela 10:

Barra de título –

Pr-Viquen

Botões de funções e botões de *status* -

Identificação - Ok

Localização na tela - Primeiro botão

Procedimento de seleção e ativação - Posicionar o cursor no botão, pressionar o botão esquerdo do mouse e soltar.