

GERALDO RANTHUM

**UM AMBIENTE DE INTERFACE PARA
VISUALIZAÇÃO DA RESPOSTA EM
UMA FERRAMENTA DE CONSULTA
A BANCO DE DADOS**

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.^ª 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 do aluno *Geraldo Ranthum*, avaliamos o trabalho intitulado "*Um Ambiente de Interface para Visualização da Resposta em uma Ferramenta de Consulta a Banco de Dados*". cuja defesa foi realizada no dia 28 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, 28 de Fevereiro de 2002.

Prof.^a Dra. Laura Sanchez Garcia
PGINF/UFPR - Orientadora

Prof.^a Dra. Maria Saleté Marcon Gomes Vaz
INF/UEPG

Prof. Dr. Marcos Sfair Sunye
DINF/UFPR

Aos meus filhos

Lucas e Larissa fontes inesgotáveis de minhas forças.

AGRADECIMENTOS

A Deus

Pela saúde e tudo que me permitiu realizar.

A minha orientadora Professora Laura

Pela motivação e esclarecimentos indispensáveis para a conclusão deste trabalho.

Aos Professores Marcos Sunye e Alexandre Direne

Pelas valiosas contribuições para este trabalho.

Aos meu amigos

A vocês, Clodis, Ezequiel, Mônica e Simone, muito obrigado pelo auxílio e pelo incentivo, ambos igualmente importantes e que nunca faltaram.

SUMÁRIO

LISTA DE FIGURAS	iii
LISTA DE QUADROS	iv
LISTA DE ABREVIATURAS E SIGLAS	v
RESUMO.....	vi
ABSTRACT	vii
1 INTRODUÇÃO	1
1.1 PROBLEMA	1
1.2 OBJETIVOS DA DISSERTAÇÃO	3
1.3 ESCOPO DA DISSERTAÇÃO	3
1.4 ESTRUTURA DA DISSERTAÇÃO	4
2 BASES TEÓRICAS	5
2.1 ABORDAGENS COGNITIVAS	5
2.2 ABORDAGENS SEMIÓTICAS	8
2.3 MODELOS E FORMALISMOS PARA A AVALIAÇÃO E O <i>DESIGN</i> DE INTERFACES.....	10
2.3.1 <i>Linguagem de Especificação da Mensagem do Designer (LEMD)</i>	11
2.3.2 <i>Semiotic Tag (STAG)</i>	14
2.4 O PROCESSO DE <i>DESIGN</i> DE INTERFACES	16
2.4.1 <i>Design de uma Linguagem Visual</i>	16
2.4.2 <i>Prototipação</i>	17
3 MODELO DE DADOS ERC+.....	20
3.1 DESCRIÇÃO DO MODELO ERC+.....	21
3.2 ÁLGEBRA ERC+.....	24
4 FERRAMENTAS DE CONSULTA A BANCO DE DADOS	31
4.1 QBD*	33

4.2	SUPER.....	35
4.3	VIQUEN.....	39
5	A INTERFACE PARA A APRESENTAÇÃO DO RESULTADO.....	43
5.1	DOMÍNIO DA APLICAÇÃO	43
5.2	PERFIL DO USUÁRIO	45
5.3	ANÁLISE DE TAREFAS.....	46
5.4	LEMD DA INTERFACE PARA VISUALIZAÇÃO DA RESPOSTA.....	47
5.5	REGRAS DE CORRELAÇÃO (MAPEAMENTO SEMÂNTICO)	47
5.6	STAG DA INTERFACE PARA VISUALIZAÇÃO DA RESPOSTA	48
5.7	A INTERFACE PROPOSTA.....	48
5.7.1	<i>Caracterização geral.....</i>	<i>48</i>
5.7.2	<i>Modos de exibição.....</i>	<i>53</i>
5.7.3	<i>Contribuições do Ambiente.....</i>	<i>59</i>
5.7.4	<i>Quadro Comparativo.....</i>	<i>60</i>
6	CONCLUSÕES E TRABALHOS FUTUROS.....	63
6.1	CONCLUSÕES	63
6.2	TRABALHOS FUTUROS	64
	REFERÊNCIAS BIBLIOGRÁFICAS	65
	APÊNDICE 1 – AUXÍLIO AO USUÁRIO	70
	APÊNDICE 2 – LEMD DA INTERFACE PROPOSTA.....	77
	APÊNDICE 3 – STAG DA INTERFACE PROPOSTA.....	81

LISTA DE FIGURAS

FIGURA 1 - AÇÕES DO USUÁRIO NA INTERAÇÃO COM O SISTEMA	7
FIGURA 2 – ETAPAS DO PROCESSO DE <i>DESIGN</i> DE INTERFACES	18
FIGURA 3 – ESQUEMA NO MODELO ERC+	23
FIGURA 4 - APRESENTAÇÃO DA RESPOSTA NO QBD*	34
FIGURA 5 - RESPOSTA NO MODO FORMULÁRIO – SUPER	37
FIGURA 6 - RESPOSTA NO MODO GRÁFICO – SUPER	38
FIGURA 7 - APRESENTAÇÃO DA RESPOSTA NO VIQUEN	41
FIGURA 8 - DIAGRAMA ERC+ DO DOMÍNIO DA APLICAÇÃO	44
FIGURA 9 – PROCESSO DE <i>DESIGN</i> DA INTERFACE	48
FIGURA 10 – JANELA INICIAL DA RESPOSTA.....	50
FIGURA 11 – JANELA PRINCIPAL DO <i>HELP</i>	53
FIGURA 12 – JANELA DA RESPOSTA NO MODO FORMULÁRIO	55
FIGURA 13 – JANELA DA RESPOSTA NO MODO GRÁFICO	57
FIGURA 14 – JANELA DA RESPOSTA NO MODO PAINÉIS	59

LISTA DE QUADROS

QUADRO 1 – AÇÕES DISPONÍVEIS AO USUÁRIO NA VISUALIZAÇÃO DA RESPOSTA	46
QUADRO 2 - REGRAS DE CORRELAÇÃO SEMÂNTICA	47
QUADRO 3 – ANÁLISE COMPARATIVA DAS FERRAMENTAS DE CONSULTA ..	61

LISTA DE ABREVIATURAS E SIGLAS

ER	ENTIDADE-RELACIONAMENTO
ES	ENGENHARIA SEMIÓTICA
ERC+	ENTIDADE-RELACIONAMENTO COMPLEXO ESTENDIDO
GUI	<i>GRAPHICAL USER INTERFACE</i>
IHC	INTERAÇÃO HUMANO-COMPUTADOR
LEMD	LINGUAGEM DE ESPECIFICAÇÃO DA MENSAGEM DO <i>DESIGNER</i>
LV	LINGUAGEM VISUAL
MD	MANIPULAÇÃO DIRETA
QBD*	<i>QUERY BY DIAGRAM</i>
QBE	<i>QUERY BY EXAMPLE</i>
QUEL	<i>QUERY LANGUAGE</i>
SGBD	SISTEMA GERENCIADOR DE BANCO DE DADOS
SQL	<i>STRUCTURED QUERY LANGUAGE</i>
STAG	<i>SEMIOTIC TASK ACTION GRAMMAR</i>
TAG	<i>TASK-ACTION GRAMMARS</i>
UCSD	<i>USER CENTERED SYSTEM DESIGN</i>
VIQUEN	<i>VISUAL QUERY ENVIRONMENT</i>
WIMP	<i>WINDOWS, ICONS, MENUS AND POINTING DEVICES</i>

RESUMO

Este trabalho apresenta um ambiente de interface para o protótipo VIQUEN, capaz de diminuir o esforço do usuário na percepção e interpretação do resultado das consultas, preservando a estrutura de objetos complexos. A dissertação discute que, do ponto de vista da Interação Humano-Computador (IHC), o processo de *design* de interfaces adequadas encontra apoio na Engenharia Cognitiva, teoria clássica na área, e na Engenharia Semiótica, que a complementa. O trabalho complementa a revisão teórica descrevendo dois formalismos que foram utilizados como sustentação teórica e orientação no desenvolvimento deste protótipo: a Linguagem de Especificação da Mensagem do Designer (LEMD) que permitiu melhorar a formulação do modelo de usabilidade da interface e a *Semiotic Tag* (STAG) como notação de auxílio na especificação dos tipos expressivos e de conteúdo. Em seguida o trabalho descreve o processo de desenvolvimento do ambiente de apresentação do resultado da consulta, baseado em princípios de *design* aplicados a linguagem visual de hipótese e apresenta um protótipo. Finalmente, discute as contribuições esperadas e os espaços para trabalhos futuros.

Palavras-chave: Interfaces, Interação Humano-Computador, Banco de Dados, LEMD, STAG

ABSTRACT

This work presents an interface environment into the VIQUEN prototype, able to reduce the effort of the user in the perception and evaluation of the result of the queries preserving the structure of the complex objects. According to the Human-Computer Interaction, the design process of interfaces gets support from the Cognitive Engineering, a classical theory in this area, and from the Semiotic Engineering which complements it. The Designer's Message Specification Language [21] and the Semiotic Tag [23] were used as a theoretical framework in the development of this prototype. The former allowed the improvement of the formulation of the interface usability model, the latter worked as an aid notation in the specification of the expressive and contents types. The work describes the development of the query result presentation, focusing design principles allied with visual language and presents a prototype. Finally discuss the contributions and the area for future works

Keywords: Interfaces, Human Computer Interaction, Databases, LEMD, STAG

1 INTRODUÇÃO

1.1 PROBLEMA

Devido à demanda crescente por informações dentro das organizações, os usuários finais de um sistema de bancos de dados devem ter a possibilidade de acessar os dados neles existentes de maneira facilitada.

Em grandes bases de dados, freqüentemente é difícil formular uma consulta suficientemente precisa para obter a exata informação desejada, uma vez que a formulação da consulta sintática e semanticamente correta requer um detalhado conhecimento do esquema, levando o usuário não familiar com o esquema a um exaustivo processo de tentativa e erro.

Surge então, um requisito bastante complexo para as interfaces de software: conciliar suas capacidades de processamento de informação extremamente poderosas e diferentes, com a capacidade do homem e do computador.

Nos últimos anos, conduzidas pelo número crescente de usuários de Sistemas Gerenciadores de Bases de Dados (SGBDs), especialmente de usuários finais, as pesquisas em interfaces para bases de dados têm motivado diversos *workshops* e publicações. Ferramentas de software têm sido apresentadas com a finalidade de facilitar o acesso destes usuários a SGBDs [3],[16],[17],[19],[20].

O aprendizado de linguagens de consulta tradicionais e a necessidade do conhecimento do esquema das bases de dados estão sendo substituídos pela abordagem de Manipulação Direta (MD) de elementos gráficos para formulação de consultas, que envolve paradigmas de representação visual, tais como formulários, diagramas e ícones [37] e a ação do usuário sobre estes, isto é, operações (tarefas) sobre as representações para interagir com a base de dados.

As interfaces visuais para bases de dados objetivam aproximar os conceitos do mundo real, percebidos pelo usuário, dos esquemas que descrevem os dados dos sistemas de informação.

O suporte necessário à representação visual e às tarefas feitas pelo usuário é dado pelo modelo de dados, que deve ser capaz de capturar os conceitos da base de dados sobre a qual a interface atua. Portanto, aplicações de bases de dados devem utilizar modelos de dados capazes de capturar os conceitos do mundo real

percebidos pelo usuário, ou seja, modelos de dados de grande poder semântico. Por outro lado, o domínio de SGBDs baseados em modelos tradicionais é evidente. Assim, há espaço para contribuições a fim de aproximá-los.

Apesar dos avanços decorrentes da utilização do paradigma da MD, a necessidade de uma melhor interação entre usuários e sistemas de banco de dados tem sido amplamente reconhecida e discutida [37]. Há consenso entre os autores em que a qualidade da interação depende principalmente das características da interface. Portanto, o ponto crucial é produzir interfaces que se apoiem em dois pontos básicos: o modelo conceitual de banco de dados e a interface propriamente dita, entendida do ponto de vista do usuário.

Um banco de dados é tipicamente definido em termos de um modelo de dados, mas os conceitos abstratos de tal modelo precisam ser expressos por meio de uma representação dentro da correspondência muitos-para-muitos existente entre o modelo de dados e as representações (visíveis ou não). A representação deve ser parte da interface, enquanto o modelo de dados deve ser parte do sistema de banco de dados. Por outro lado, há modelos de dados, também chamados modelos semânticos, que transferem mais informações para o usuário, oferecendo ricos mecanismos para estruturar os dados contidos no banco de dados e mostrando mais relacionamentos entre eles. Além disto, uma representação deve ser efetiva, mas não pode adicionar semântica aos dados.

Várias extensões a modelos tradicionais e ferramentas associadas têm sido propostas com o objetivo de aumentar o poder semântico de modelos tradicionais. Entre elas estão o modelo ERC+ [37] e o ambiente VIQUEN (*VISUAL QUERY ENVIRONMENT*) [19], construído para este modelo.

O VIQUEN [19] é um ambiente interativo para consulta visual e extração de esquemas de uma base de dados construído com o objetivo de atingir a categoria de usuários finais, os quais buscam freqüentemente informações armazenadas em sistemas de banco de dados, mas desconhecem a sintaxe de linguagens de consulta, construídas tipicamente para uso de especialistas. Este ambiente recupera um esquema de base de dados construído sobre o modelo relacional, de pouco poder de expressão do ponto de vista do usuário e traduz para o modelo ERC+ [37].

Apesar de poder ser caracterizado como um ambiente de consulta visual, o VIQUEN não foi desenvolvido tendo-se a interface-usuário como foco principal.

Do ponto de vista da Interação Humano-Computador (IHC), o processo de *design* de interfaces adequadas encontra apoio na Engenharia Cognitiva [28], teoria clássica na área, e na Engenharia Semiótica [12], que a complementa.

1.2 OBJETIVOS DA DISSERTAÇÃO

Partindo do contexto recém caracterizado, este trabalho se propõe, a desenvolver um ambiente de interface para o protótipo VIQUEN [19], capaz de otimizar, do ponto de vista do usuário, o processo de apresentação e avaliação do resultado das consultas preservando a forma de objetos complexos.

1.3 ESCOPO DA DISSERTAÇÃO

Embora o processo de consulta a base de dados seja executado em três fases (seleção do sub-esquema, formação do predicado e apresentação do resultado da consulta) este trabalho se restringirá à terceira delas, ou seja, das formas como o usuário poderá visualizar os dados resultantes da seleção do sub-esquema [6] e da especificação do predicado [7], outros dois trabalhos que abordam as fases iniciais do processo de consulta.

O enfoque principal deste trabalho envolve o *design* de interfaces, a interação usuário-sistema e a aquisição do conhecimento necessário para utilizar um sistema computacional – o protótipo VIQUEN [19]. Embora as atividades cognitivas sejam fundamentais no processo de interação, a perspectiva que é adotada neste trabalho desloca o foco dos modelos cognitivos para os processos comunicativos e semióticos que envolvem a interação, interpretação, conhecimento e aprendizado.

Este trabalho considera interfaces de usuário gráficas *GUI* (*Graphical User Interface*) no estilo *WIMP*¹ (*Windows, Ícons, Menus e Point Devices*). Também considera aplicações de software destinadas a computadores pessoais, com usuário único, que tenha algum contato com interfaces de *software* padrão *WIMP*.

¹ O estilo de interação *WIMP*, um acrônimo em inglês[31],[36] para Janelas, Ícones, Menus e Apontadores, permite a interação por meio de componentes de interação virtuais denominados *widgets*. Nas interfaces *WIMP* é possível encontrar os estilos por menus, manipulação direta, preenchimento de formulário e linguagem de comandos.

1.4 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação esta estruturada como segue:

No capítulo um é descrito o problema clássico de interfaces para banco de dados. É relatado, também, que, na área de Sistemas Gerenciadores de Banco de Dados, as pesquisas e publicações sobre o tema estão em fase emergente, especialmente na área de interfaces visuais, as quais ainda não alcançaram resultados completamente satisfatórios. Em seguida, apresentou os objetivos, o escopo e a estrutura da dissertação.

O capítulo dois faz um embasamento conceitual na área de Interação Humano-Computador, descrevendo as abordagens teóricas dominantes nesta área, a Engenharia Cognitiva [28] e a Engenharia Semiótica (ES) [12] e dois métodos semióticos para a avaliação e *design* de interfaces: a Linguagem de Especificação da Mensagem do *Designer* (LEMD) [21] e a *Semiotic Tag* (STAG) [23].

O capítulo três faz uma breve revisão sobre modelos de dados e descreve o modelo ERC+ [37], trazendo um exemplo de uso. Mostra também que operações com bases de dados relacionais podem ainda ser feitas no modelo ERC+, semanticamente mais rico.

O capítulo quatro descreve algumas ferramentas de interfaces visuais para consulta a banco de dados, destacando alguns protótipos estudados e avaliados por pesquisadores. Descreve aspectos gerais de cada ferramenta e faz a avaliação dos aspectos relacionados à fase de apresentação do resultado.

O capítulo cinco apresenta a solução do ambiente de interface para a visualização do resultado das consultas do usuário para o protótipo VIQUEN [19] constituída a partir do embasamento teórico fornecido pela abordagem da Engenharia Semiótica

O capítulo seis apresenta as conclusões da dissertação abordando as contribuições esperadas deste trabalho e o espaço para trabalhos futuros.

2 BASES TEÓRICAS

Neste capítulo é feito o embasamento conceitual, descrevendo as abordagens teóricas dominantes na área de Interação Humano-Computador. Descreve também a Engenharia Cognitiva e a Engenharia Semiótica e dois métodos semióticos para a avaliação e *design* de interfaces: a LEMD [21] e a STAG [23].

Uma abordagem para IHC envolve o conjunto de modelos e teorias para o estudo da interação usuário-sistema, que adotam fundamentos e perspectivas comuns, dos quais se derivam princípios, métodos, técnicas e ferramentas práticas para o *design* de interfaces.

2.1 ABORDAGENS COGNITIVAS

As abordagens dominantes que têm caracterizado IHC são as cognitivas [31]. Os objetivos desta abordagem envolvem a aplicação de estudos de Psicologia Cognitiva, Ciência Cognitiva e Inteligência Artificial para compreensão das capacidades e limitações da mente dos usuários.

A estratégia das abordagens cognitivas para o apoio ao *design* de sistemas interativos consiste na elaboração de modelos cognitivos genéricos que permitam aos *designers* entender os processos cognitivos humanos envolvidos na interação e realizar experimentos ou previsões com estes modelos. A idéia básica é que modelos cognitivos que descrevem os processos e estruturas mentais (por exemplo, recordação, interpretação, planejamento e aprendizado) podem indicar para pesquisadores e *designers* quais as propriedades que os modelos de interação devem ter, de maneira que a interação possa ocorrer mais facilmente pelos usuários. Como estas abordagens adotam uma perspectiva centrada nos aspectos cognitivos do usuário, o *design* feito com base nelas é chamado de “Projeto de Sistemas Centrado no Usuário” (*User Centered System Design – UCSD*). Uma das teorias mais conhecidas de *design* centrado no usuário é a Engenharia Cognitiva apresentada por Norman [28]. Norman considera que o *designer* inicialmente cria o seu modelo mental do sistema, chamado “modelo de *design*”, com base nos modelos do usuário e da tarefa. A implementação do *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.

Segundo Norman, as pessoas interagem com o sistema a partir da formulação de um modelo mental do sistema. Este modelo permite ao usuário mapear suas metas em comandos e funções do sistema. Segundo esta proposta, o papel do *designer* está em desenvolver uma interface que permita ao usuário, durante o processo de interação, adquirir um modelo mental correspondente ao modelo do sistema [28]. Norman considera que este modelo mental do usuário é adquirido durante a interação com a “imagem do sistema” - a interface, o sistema de ajuda e toda a documentação do sistema.

Assim, segundo a Engenharia Cognitiva, a meta do *designer* é desenvolver um sistema que permita ao usuário, durante o processo de interação, criar um modelo mental consistente com o modelo projetado pelo *designer*. Para que isto seja possível, Norman argumenta que o *designer* precisa entender o processo por meio do qual o usuário interage com a interface do sistema e propõe a teoria da ação. A **teoria da ação** [28] define que a interação usuário-sistema é desempenhada num ciclo-de-ação com seis etapas e dois “golfs” a serem atravessados. Um deles é o **golfo da execução** e envolve as etapas de formulação da intenção, especificação da seqüência de ações e atividade física de execução. O outro é o **golfo da avaliação** e deve ser atravessado pelas etapas de percepção, interpretação e avaliação da meta, conforme mostrados na FIGURA 1.

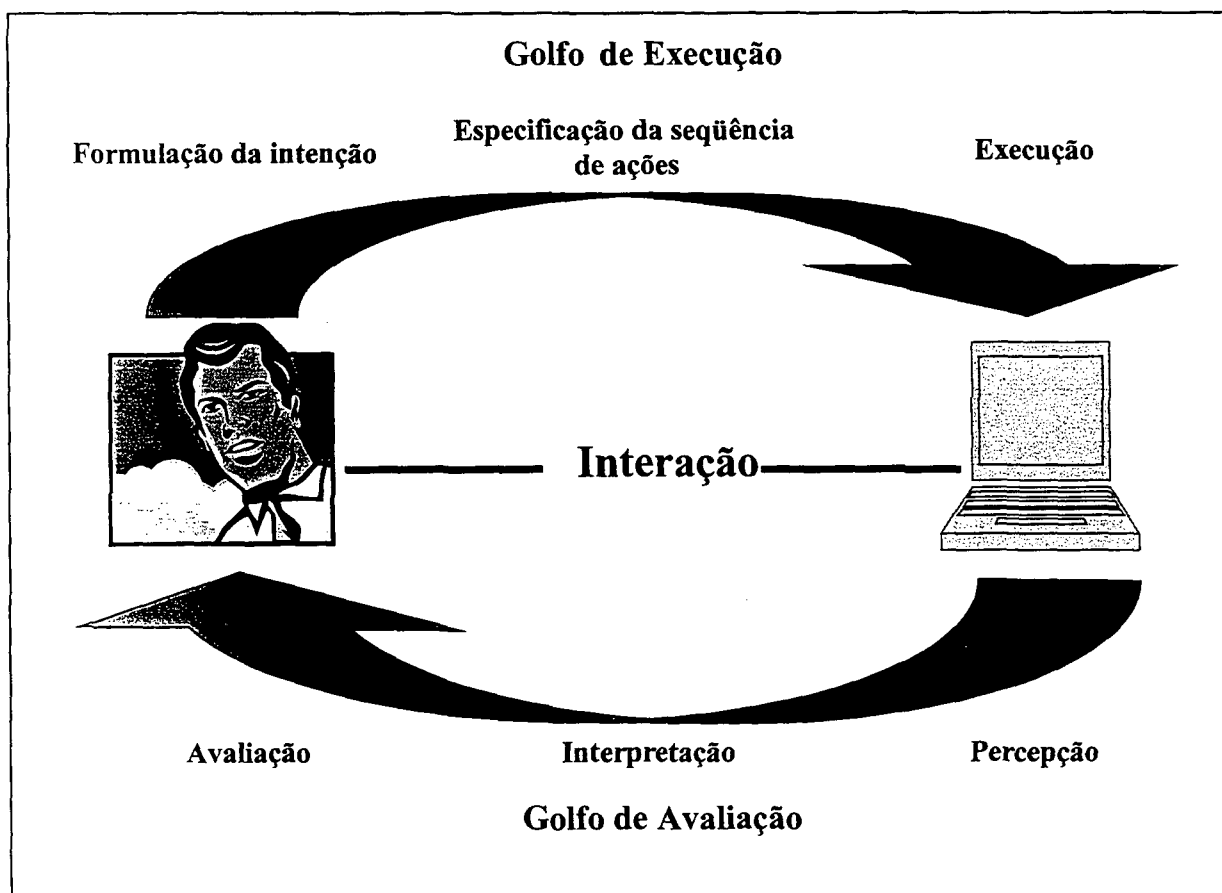


FIGURA 1 - AÇÕES DO USUÁRIO NA INTERAÇÃO COM O SISTEMA, ADAPTAÇÃO DE [28]

O *designer* pode ajudar o usuário a atravessar estes golfos diminuindo-os. Para isto ele deve definir quais são as ações e estruturas mais adequadas para comandar as funções do sistema, escolher os elementos de interface que melhor comunicam a informação desejada, optar por *feedbacks* significativos, dentre outras escolhas de *design*.

Assim, quanto mais próxima da tarefa e das necessidades do usuário for a linguagem de interface oferecida pelo *designer*, menos esforço cognitivo o usuário terá para atingir seus objetivos. Com o objetivo de aproximar a interface do usuário, retirando deste a necessidade de realizar esforço cognitivo na direção da mesma (cruzar os golfos de execução e avaliação), é preciso considerar dois conceitos no projeto.

A Distância Semântica [28] expressa se todas as funcionalidades do *software* estão representadas na interface, permitindo que o usuário tenha acesso a todo o seu potencial. No golfo de execução, a distância semântica reflete quanto da

estrutura requerida é expressa pelo sistema e quanto pelo usuário. A distância semântica será maior, quando o usuário precisar realizar esforço cognitivo na elaboração da solicitação da tarefa. No golfo de avaliação, a distância semântica será grande, quando o usuário precisar traduzir a saída em expressões compatíveis como a meta a ser atingida.

A Distância Articulatória [28] demonstra se todas as operações do software estão representadas de maneira natural para o usuário e se esta representação foi feita em uma linguagem (textual ou visual) clara. Interfaces de manipulação direta (comandos e funções representadas por ícones), geralmente, têm uma forma de expressão mais próxima do significado, propiciando com isto uma distância articulatória próxima de zero (0), tanto na entrada (golfo de execução) quanto na saída (golfo de avaliação).

2.2 ABORDAGENS SEMIÓTICAS

A Semiótica é o campo de estudos dos signos, dos sistemas de signos e dos processos envolvidos - semiose, comunicação, significação, interpretação, e outros. “Um signo, ou *representâmen*, é aquilo que, sob certo aspecto ou modo, representa algo para alguém” [21].

As abordagens semióticas para IHC se caracterizam principalmente pela aplicação de teoria semiótica como fundamentação teórica para explicações de fenômenos relativos à interação usuário-sistema e à sua aplicação no *design* de interfaces de usuários.

O foco de interesse não está diretamente centrado em atividades mentais dos usuários, mas no papel que sistemas de significação e comunicação exercem sobre estas atividades cognitivas. Ao adotar a semiótica como uma fundamentação para IHC pode-se trazer não apenas novos conceitos e teorias, mas diferentes tradições e perspectivas filosóficas.

ANDERSEN [2] foi um dos primeiros pesquisadores a introduzir fundamentos de teoria semiótica em computação propondo uma *Teoria da Semiótica Computacional*. A visão de Andersen é de que sistemas computacionais devem ser vistos como mídia ou sistema semiótico. Ele se refere à mídia não no sentido comum de rádio ou televisão, mas como um meio extremamente flexível e polimórfico. Foi a

partir da visão do Andersen, que se passou a considerar a perspectiva de que *designers* são emissores de mensagens através da mídia computacional.

O *framework* proposto por Andersen apresenta vantagens para análise de sistemas computacionais, entretanto não discute aspectos de produção e interpretação dos signos, o que torna a sua aplicação no *design* de interfaces e aplicações bastante limitadas.

A Engenharia Semiótica[13], é uma nova abordagem para IHC fundamentada na teoria semiótica, na qual a interação usuário-sistema é um processo de metacomunicação onde o usuário exerce o duplo papel de interagir com o sistema e interpretar uma mensagem enviada pelo *designer* [12]. O conteúdo de tal mensagem é o modelo de usabilidade da aplicação, tendo como características:

- o processo de interação e *design* de interfaces são considerados a partir desta perspectiva semiótica;
- as aplicações são consideradas como signos.

A Engenharia Semiótica apresenta uma perspectiva específica para o *design* de interfaces de usuário, na qual as interfaces de usuários são artefatos de metacomunicação, uma vez que eles não são apenas um sistema de comunicação usuário-aplicação, mas também uma mídia que comunica para o usuário a solução de *design* - a funcionalidade e o sistema de interação - formulada pelo *designer*. Visto por esta perspectiva, o *design* de interfaces envolve não apenas a concepção do sistema de interação, mas a *comunicação* deste modelo da aplicação, de maneira a revelar para o usuário o espectro de usabilidade da mesma.

Na Engenharia Semiótica o foco está na comunicação interpessoal entre o *designer* e os usuários. Sistemas computacionais oferecem meios de se criar um artefato *ativo* onde o sistema de interação e de funcionalidade da aplicação podem ser ensinados aos usuários. O argumento principal da Engenharia Semiótica é que o sistema de interação e a funcionalidade da aplicação devem ser parte da mensagem unidirecional e indireta enviada através da coleção de signos da interface. O *designer* passa a ter um papel fundamental ao se utilizar da interface para se comunicar com o usuário.

O *design* de interfaces deve ter como resultado interfaces que respondam a 2 perguntas do usuário: o que posso fazer com o sistema? – a funcionalidade,

especificada em termos de signos do domínio e de funções aplicadas, e como devo interagir para consegui-lo? – o modelo de interação, especificado em termos de comandos, interações básicas e visualizações [13].

Para auxiliar ao usuário nesta direção o *designer* precisa um bom projeto de comunicação e traduzi-lo consistentemente na interface. Como complementação deve proporcionar manuais, sistemas de auxílio ou tutoriais que servirão de apoio neste processo.

2.3 MODELOS E FORMALISMOS PARA A AVALIAÇÃO E O DESIGN DE INTERFACES

Nesta seção são descritos dois formalismos que servirão como sustentação teórica e orientarão a construção do protótipo de interface para a visualização da resposta de consultas feitas à base de dados.

A mensagem do *designer* deve ser utilizada para comunicar o modelo de usabilidade do sistema. A *Linguagem de Especificação da Mensagem do Designer* [21] tem como objetivo apoiá-lo na formulação da mensagem sobre o modelo de usabilidade. Ela permite especificar os *signos do domínio*, *as funções aplicadas*, *as visualizações* e *os comandos* como sendo mensagens enviadas pelo *designer*. As sentenças da linguagem descrevem as *mensagens de interação* que compõem a mensagem global.

Um dos formalismos mais conhecidos para modelos de gramáticas de ação, pois permite a elaboração de linguagens de comandos a partir de modelos de tarefas, é a *Task-Action Grammar (TAG)* *in apud* [23] é uma técnica que consiste em identificar tarefas simples, representá-las de forma categorizada através de regras de produção. O objetivo é facilitar o aprendizado e melhorar o desempenho do usuário. Assim, a ênfase está em resolver os problemas relativos à consistência. A TAG é uma meta-linguagem que é um modelo de representação mental de linguagens de tarefas e é, portanto, um modelo de competência, sendo um de seus objetivos refletir a estrutura de raciocínio do usuário, em função da tarefa [23]. Este formalismo é complementado, na visão da Engenharia Semiótica, pela STAG (*Semiotic Tag*)

2.3.1 Linguagem de Especificação da Mensagem do *Designer* (LEMD)

As interfaces podem ser entendidas como linguagens, na medida em que expressam elementos que devem ser usados em um processo de comunicação. A Linguagem de Especificação da Mensagem do *Designer* (LEMD)[21] está baseada nas principais idéias da Engenharia Semiótica e tem por objetivo específico auxiliar na apresentação da mensagem do *designer* que comunica aos usuários o modelo de interação e de funcionalidade do sistema, denominado **modelo de usabilidade**. O modelo de funcionalidade determina **o que** o usuário pode fazer, enquanto o modelo de interação determina **como** ele pode utilizar o sistema.

A contribuição da LEMD para a ES consiste na possibilidade de uma especificação da interface em nível abstrato, permitindo a representação de associações entre intenções de comunicação do *designer* e expressões de interface utilizadas para este fim, como, por exemplo, os *widgets* presentes nas ferramentas de *GUIs*, detalhados em [14].

A LEMD permite uma melhor formulação do modelo de usabilidade a ser comunicado. Ela fornece ao *designer* recursos para a estruturação da mensagem que revelam as distinções conceituais do modelo de usabilidade. As estruturas possibilitam, ainda, que tenhamos um mapeamento entre elas e os tipos expressivos, ou seja, as estruturas de interação (*widgets*) presentes na interface.

As vantagens de se utilizar a LEMD consistem em que, na medida em que ela reflete um modelo global de comunicação humana através de artefatos de meta-comunicação (as interfaces do sistema), estimula o *designer* de interfaces 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 [21].

Através 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 *designer*, e, até, representar seqüências completas de passos que o usuário deve executar para atingir uma determinada meta do seu escopo de trabalho [32].

Desta forma, a LEMD permite estruturar os diversos tipos de mensagens que o *designer* necessita enviar para comunicar ao 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 *designer* especificar as diversas outras mensagens que auxiliam no processo de utilização e aprendizado do sistema [21].

A LEMD diferencia diversos tipos de mensagens [21],[32] as quais são descritas a seguir:

- 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 e o que o usuário deve fazer para controlá-las (*View State-of <application- function>*);
- 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 seqüência (*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 *designer*. Corresponde à estrutura de navegação entre telas (*Show Command-message*);

- Mensagens de meta-comunicação direta: permitem ao *designer* 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.

Proposta inicialmente como linguagem de auxílio ao *design*, a LEMD permite [21], [32] analisar as intenções comunicativas do *designer* e pode ser a base para a avaliação de interfaces existentes. Esta avaliação é motivada pela necessidade de localizar, da forma mais precisa possível, partes da interface onde as mensagens do *designer* não refletem o modelo de usabilidade, e seu procedimento, conforme [32], 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 *designer*, verificar o grau de dificuldade na sua detecção e compará-lo ao modelo intencionado. Ainda, o processo de decodificação da interface pelo avaliador, proposto em [12], 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 *layout* e da própria dinâmica dos elementos. Durante o processo de interpretação da interface e descrição segundo a 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 *designer* expressa através do código visual (*widgets* e *layout*).

Estes pontos são assinalados na descrição segundo a 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 *designer*-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 *designer*. 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 *designer*, na impossibilidade deste, documentação *on-line* e manuais de usuário também são fontes válidas [21].

Do confronto entre o modelo de usabilidade descrito em LEMD (meta-linguagem) e as fontes alternativas, pode-se detectar deficiências na comunicação *designer*-usuário, onde, por exemplo, o usuário entende que deve utilizar um modo de interação com o sistema diferente do que o *designer* intencionara. Tais pontos da interface sugerem que o usuário não estaria recebendo convenientemente a mensagem do *designer* [13].

A execução da avaliação proposta por [32], utilizando a LEMD, em duas etapas (interpretação da interface e documentação da mensagem do *designer*, e posterior comparação com o modelo de usabilidade), mostra-se interessante por minimizar a predisposição do avaliador na interpretação da mensagem do *designer* presente na interface.

2.3.2 Semiotic Tag (STAG)

A TAG será utilizada neste trabalho através da STAG (*Semiotic TAG*) [22]. A STAG é uma notação para auxílio à especificação de Linguagem Visual (LV) por MD, que orienta a sistematização de escolhas expressivas. Ela foi proposta como uma extensão da TAG (*Task-Action Grammar*). Parte dos problemas da TAG decorrem especificamente de sua aplicação à LV, outra parte é intrínseca à própria estrutura da gramática. A proposta da STAG é que a sua notação complemente a TAG, superando alguns pontos de deficiência, para aplicação às LVs.

Os requisitos para a STAG foram estabelecidos considerando os seguintes aspectos: (i) a notação deve estar apoiada em uma base teórica sólida, neste caso instanciada em um enfoque semiótico, e (ii) deve atender às necessidades específicas das LVs. Na proposta da STAG entende-se que não é o caso de, o

designer da LV ser um semioticista. Por isto, a notação incorpora implicitamente alguns princípios básicos da Semiótica, de forma que estes princípios sejam automaticamente respeitados, pela simples utilização da mesma.

A STAG deve auxiliar o *designer* a definir a linguagem de interface dentro da proposta semiótica, observando orientações específicas para LVs que favoreçam o alcance de um bom resultado. A STAG deverá atender às necessidades de projeto e análise. Quando utilizada para projeto, ela estará modelando a concepção do *designer* sobre o problema. Quando utilizada para análise, ela estará registrando o processo sógnico do usuário, ou seja, seus interpretantes, nem sempre previstos pelo *designer*.

A STAG é uma notação de auxílio, ela não se propõe a determinar o que deve ser usado. Mesmo sendo implementada por meio de um aplicativo, ela não prescreve soluções para o *designer*. Ela apenas sugere soluções mais adequadas, de acordo com suas heurísticas. A decisão final a respeito da LV será sempre do *designer*.

A STAG tem quatro componentes: (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 que chamaremos de gramática, para distingui-lo do mapeamento. Este conjunto de componentes será utilizado dentro de uma estrutura hierárquica, que permite gerar uma linguagem com seções paralelas, entre as quais o usuário poderá alternar durante o uso da linguagem.

A gramática é gerativa. A saída da gramática é uma lista de tipos e instâncias do conteúdo utilizado pelo usuário e sistema durante a execução de uma tarefa. Estes tipos e instâncias do conteúdo são mapeados em tipos e instâncias da expressão. De forma semelhante à TAG, a STAG tem capacidades estritamente livres de contexto, apesar de uma certa parametrização dos constituintes gramaticais. Cada regra ordinária pode ser expandida em regras de re-escrita de níveis simples (livres de contexto), simplesmente associando valores às características em todos os caminhos possíveis.

De modo geral, a autora [23] avalia que o grande apoio que a notação oferece é “forçar o *designer* a especificar os tipos expressivos e do conteúdo, do usuário e

sistema”. Mesmo que a notação não dê conta de controlar automaticamente as possíveis inadequações, o simples fato de o *designer* ser levado a pensar a interface como uma linguagem e a especificar os elementos básicos desta linguagem fará com que ele próprio se dê conta de boa parte dos problemas evitando-os no projeto [23]. O enfoque sistemático da linguagem de interface é o grande ganho do *designer*, no uso da notação.

2.4 O PROCESSO DE *DESIGN* DE INTERFACES

Design é a atividade intelectual de conceber e descrever um produto a partir dos requisitos de seus potenciais usuários [13]. Esta atividade requer técnicas e ferramentas adequadas, aliadas à criatividade, ao talento e à experiência do *designer*. No desenvolvimento do ambiente de apresentação do resultado da consulta, visando expandir as potencialidades do ambiente VIQUEN, foram utilizados princípios de *design* aliado a linguagens visuais e foi desenvolvido um protótipo.

2.4.1 *Design* de uma Linguagem Visual

A 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 ao usuário um conjunto de estratégias metodológicas para diminuir seu esforço cognitivo durante a interação com o sistema.

O usuário necessita potencialmente 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 dificuldades eventualmente presentes no momento que este está trabalhando em um domínio desconhecido. Dentre tais dificuldades podem ser citadas: descobrimento de informações implícitas, análise de esquemas detalhados, necessidade de armazenar informações pré-requeridas, entre outras. [3]

Sendo assim, ao invés de apresentar uma exaustiva lista de tuplas as quais satisfazem às restrições impostas pela consulta, um retorno que descreva a resposta de maneira visual é preferível e pode realmente conter mais informação útil ao usuário da aplicação.

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 *designer* do sistema [13]. Assim sendo, o sistema proposto deve ser construído para facilitar o desempenho do usuário no que diz respeito à realização no sistema das tarefas que ele tem em mente.

Quando se faz uso de signos, ícones e representações gráficas em geral, está sendo utilizada uma linguagem visual, que deve seguir um formalismo, para que o usuário tenha bem definidas as regras de formação desta linguagem mesmo quando realizada por meio de elementos gráficos [23].

Com base na análise das ferramentas acadêmicas Pasta-3, QBD* e SUPER e, ainda, na avaliação semiótica do protótipo VIQUEN, todas focando a etapa de apresentação do resultado da consulta, a linguagem visual desenvolvida determinará a ordem e utilização dos *widgets*, a sintaxe das frases utilizadas na ajuda, as linhas de cabeçalho e o corpo das telas.

2.4.2 Prototipação

O produto concebido em uma atividade de *design* precisa ser apresentado por meio de um protótipo e/ou de uma especificação. A prototipação consiste na descrição do que foi concebido, permitindo uma descrição e avaliação mais concreta do produto no contexto de utilização. A especificação consiste na descrição abstrata, rigorosa, idealmente correta e completa do produto, utilizando uma notação ou linguagem adequadas.

O processo de *design* de interfaces de usuário é uma atividade que requer análise dos requisitos dos usuários, *design*, prototipação da interface e avaliação da utilização do protótipo pelos usuários. Diversos modelos de desenvolvimento de *software* baseados em prototipação argumentam que este processo deve ser realizado de forma cíclica, isto é, a avaliação deve levar a um novo *design* e o resultado deste deve ser posteriormente avaliado [21]. O processo de *design* está representado na FIGURA 2

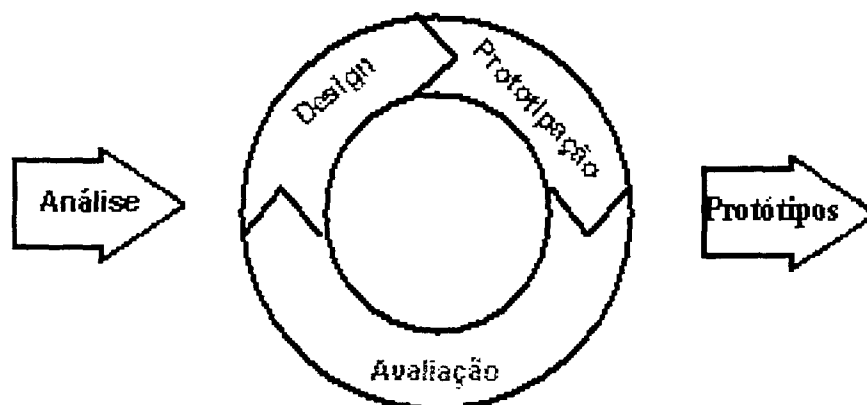


FIGURA 2 – ETAPAS DO PROCESSO DE DESIGN DE INTERFACES [21]

O *design* envolve a concepção e a especificação da interface. A prototipação pode ser realizada utilizando-se ferramentas baseadas em *widgets*. Em alguns métodos, as etapas de *design* e prototipação são desempenhadas conjuntamente, isto é, a interface é concebida e concretizada diretamente na construção de um protótipo. Em outros, o *design* pode ser especificado numa notação apropriada, antes da construção de um protótipo como descrito em [21].

Após uma prototipação inicial ter sido feita, a sua avaliação deve ser realizada por meio da elaboração de cenários de uso, análise de tarefas com o protótipo e testes de usabilidade, para que se possa realizar os ajustes e se chegar a um protótipo final. A avaliação da usabilidade requer a definição de critérios específicos para a aplicação de testes sistemáticos.

Uma interface gráfica é composta de diversos componentes visuais interativos, chamados *widgets*. Para cada ambiente gráfico, geralmente existe uma ou mais bibliotecas de *widgets*. Uma biblioteca define padrões de aparência e comportamento da interface chamados comumente de *look and feel* de uma interface gráfica. Para implementar interfaces gráficas utilizando *widgets*, utilizamos *toolkits* [12].

Um *toolkit* pode ser utilizado para mais de uma plataforma, ou para uma plataforma específica. Um *toolkit* oferece uma ou mais bibliotecas de *widgets*. Ferramentas de alto nível para apoio à construção de interfaces facilitam a utilização de *toolkits* e sistemas de janelas para a construção de interfaces gráficas. Geralmente são ferramentas de implementação, que permitem construir desde

protótipos a sistemas inteiros. Para o ambiente Windows, por exemplo, o Microsoft Visual Basic e o Borland Delphi² podem ser considerados ferramentas de apoio à construção de interfaces gráficas.

É importante notar que a utilização de ferramentas de apoio à construção de interfaces não é o suficiente para garantir a qualidade da interface resultante. Elas devem ser utilizadas após terem sido feitas análises e modelagens de usuários, de tarefas e de comunicação da aplicação, ou seja, após o *designer* ter toda a informação necessária para tomar decisões de *design* acertadas.

Na abordagem da Engenharia Semiótica, a mensagem do *designer* contempla tanto a funcionalidade quanto o modelo de interação. A interface é, portanto, responsável por fazer o usuário ter condições de interagir com a funcionalidade do sistema. O *design* da interface de usuário depende da especificação do modelo de interação (que visa descrever de forma abstrata e precisa como o usuário pode interagir com o sistema independente de quais dispositivos de interação ou *widgets* ele vai utilizar) e do modelo de funcionalidade (quais funções o sistema deve oferecer para o usuário) do sistema. O *design* da interface de apresentação da resposta a consultas objeto deste trabalho seguirá esta perspectiva.

² O Borland Delphi será a ferramenta de apoio à construção de interfaces utilizada no desenvolvimento do protótipo apresentado neste trabalho.

3 MODELO DE DADOS ERC+

Neste capítulo será feita uma breve revisão sobre modelos de dados e descrito o modelo ERC+ trazendo um exemplo deste. Será mostrado também que operações em bases de dados relacionais podem ser feitas neste modelo que é semanticamente mais rico.

Segundo NAVATHE e ELMASRI [26] pode-se definir modelo de dados como sendo um conjunto de conceitos que podem ser usados para descrever a estrutura de um banco de dados. Pode-se categorizar estes modelos de acordo com os conceitos que utilizam para descrever a estrutura de um banco de dados.

O modelo Entidade-relacionamento foi desenvolvido em 1976, por Peter Chen, como uma ferramenta para a modelagem de dados. O Modelo ER trabalha com três conceitos básicos:

- entidade (objetos do negócio);
- relacionamentos (associação entre os objetos);
- atributos (propriedades dos objetos e dos relacionamentos).

O modelo Entidade-relacionamento, 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. Há um vasto legado de informação armazenado neste modelo. Os modelos de dados existentes fazem uso de conceitos similares para representar a informação, dos quais podem ser citados:

- 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 somente “intensional”.

Neste sentido, pode-se caracterizar as respostas às consultas como extensionais e intensionais. Uma resposta extensional para uma dada consulta é caracterizada por um conjunto de valores de dados presentes no banco de dados, que satisfazem às condições da consulta. Uma resposta intensional é um conjunto de informações que caracterizam a resposta extensional obtida, proporcionando informações de maior utilidade em relação à composição do conjunto de valores recuperados.

3.1 DESCRIÇÃO DO MODELO ERC+

Várias extensões a modelos tradicionais têm sido propostas com o objetivo de aumentar o seu poder semântico. Entre elas está o modelo Entidade-relacionamento Complexo (ERC+) [37], que é um modelo Entidade-relacionamento estendido, que adiciona ao paradigma Entidade-relacionamento conceitos do modelo comportamental da orientação a objetos e outros semânticos como relacionamento genérico. O modelo ERC+ é um modelo ER estendido que foi definido para apoiar a descrição de objeto complexo e sua manipulação. Este modelo é chamado ERC+ devido ao seu significado: ER para Objetos Complexos (o + denota o enriquecimento do modelo ERC básico).

A modelagem do objeto complexo e seu gerenciamento são metas principais dos modelos de dados de hoje. Por meio de um objeto complexo, queremos dizer 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 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.

Com o modelo ERC+, é possível implementar aplicações de consultas para bases de dados que permitam ao usuário: visualizar e recuperar a informação de esquemas sobre um modelo de dados semanticamente mais rico, o modelo ERC+,

apesar de a informação residir em bases de dados relacionais, e expressar suas consultas sobre este modelo. A seguir é descrito o modelo ERC+

A estrutura de uma entidade consiste em um conjunto de um ou mais atributos. Os relacionamentos podem unir qualquer número de entidade e são cíclicos se a mesma entidade participa, mais de uma vez, no relacionamento.

Um papel é associado a cada participação de uma entidade em um relacionamento. Este papel é caracterizado por suas cardinalidades, especificadas como 0-1, 0-n, 1-1 ou 1-n de acordo com o vínculo da entidade para o relacionamento. Entidades e relacionamentos podem ter zero, um ou mais conjuntos de atributos que servem como identificadores.

Atributos podem ser: obrigatórios ou opcionais, monovalorados ou multivalorados e simples ou complexos. Um atributo identificador indica que o valor deste atributo é distinto para cada instância desta entidade, sendo que o mesmo é utilizado como identificação da instância.

Duas generalizações são apoiadas 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.

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.

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. Na FIGURA 3, é mostrado um esquema ERC+ juntamente com a notação utilizada.

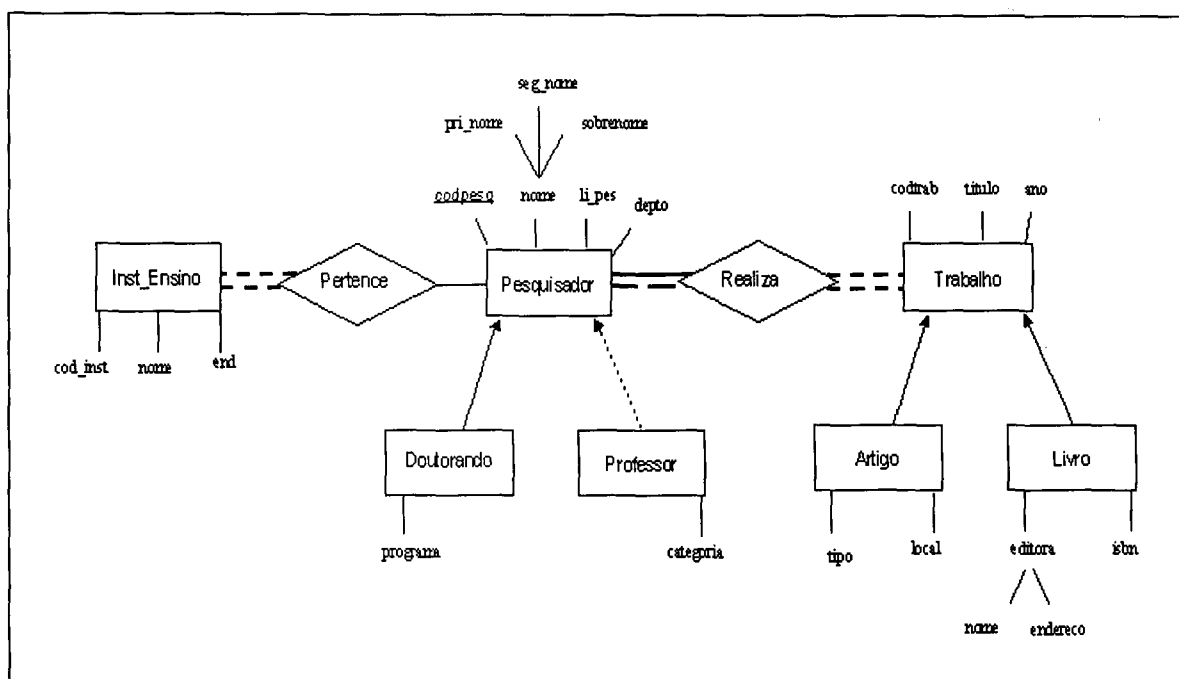


FIGURA 3 – ESQUEMA NO MODELO ERC+ [35]

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 mandatório, 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 mandatória 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*.

O modelo ERC+ é o modelo subjacente ao protótipo VIQUEN [19].

A álgebra ERC+ [37] foi construída para manipular todos os conceitos do modelo ERC+, desta forma ela foi utilizada como origem do mapeamento para SQL, visto que o SGBD subjacente, adotado para o protótipo VIQUEN [19], tem como alvo esquemas relacionais.

A álgebra ERC+ é um conjunto de operadores primitivos que podem ser combinados em qualquer ordem dentro de expressões, de tal forma que qualquer consulta sobre um base de dados ERC+ possa ser satisfeita por uma expressão algébrica apropriada.

A propriedade de fechamento³ e o poder de expressão da álgebra permitem cinco pressupostos básicos:

- operandos e resultados são do tipo entidade. Cada operador é projetado para manipular um (ou mais) tipo(s) entidade e para construir o resultado como um tipo entidade derivado. Ainda o resultado pode servir como um operando para um operador subsequente.
- Tipos entidade resultantes são complementados com tipos relacionamentos, generalização e ligação de conjunção derivados dos operandos. Este procedimento permite que o resultado das expressões sejam manipulados normalmente como tipos entidades. Também expressões algébricas de qualquer complexidade podem ser definida. A álgebra ERC+ é completa.
- a álgebra preserva objeto. Cada ocorrência do resultado é derivada dos operandos, que são entidades, do seguinte modo:
 - seu identificador de objeto (oid) é idêntico ao oid da entidade operando do qual ele veio.
 - seu valor é derivado dos valores do(s) operando(s).
- seus relacionamentos são derivados daqueles que ligam os operandos;
- suas ligações de generalização e conjunção são derivadas daquelas que envolvem os operandos;

Devido à característica de o modelo suportar objetos complexos, os operadores constroem seus tipos de entidades resultantes como objetos complexos.

³ A álgebra suporta a semântica do modelo [37].

Em particular os operadores *product*⁴, *relationship-join*⁵ e *identity-join*⁶ (usados para transformar diferentes tipos de entidades dentro de um simples objeto) usam estruturas de atributos complexo para inserir a informação dos outros operandos dentro de um operando principal, do tipo entidade. *Product*, *relationship-join* são operadores que retornam estruturas não planares⁷. A ocorrência dos outros operandos são transformadas dentro de um atributo complexo multivalorado.

Por exemplo, a junção relacional através de um relacionamento cria um novo tipo entidade, cujo esquema compreende o esquema original acrescido de um atributo complexo, agrupando todos seus atributos.

Esta particularidade da álgebra ERC+ tem duas vantagens principais:

- o preenchimento de todos os requisitos do usuário para produzir estruturas não planares que mostrem para cada ocorrência da entidade principal todas as ocorrências de entidades secundárias ligadas a ela.
- operadores binários com um operando principal (produto e junção-relacional) preservam os objetos: os oids dos resultados são derivados (são iguais) daqueles do operando principal. Com estruturas planares esta condição não seria possível.

Herança através de ligações *is-a* (“é um”) e *may-be-a* (“pode-ser-um”) é efetuada de duas maneiras. A álgebra oferece um operador, o *identity-join*, o qual permite ao usuário agrupar dois tipos entidades através de ligações *is-a* ou *may-be-a* dentro de um tipo entidade simples. Todos os demais operadores usam somente suas próprias propriedades (não herdam) de seus operandos.

A álgebra inclui nove operadores primitivos. Operadores derivados podem também ser definidos para facilitar a elaboração de consultas envolvendo vários operadores. A seguir serão apresentados brevemente os operadores.

Operador *selection*⁸, é similar ao da álgebra relacional.

A operação: $E = \text{select} [\text{predicado}] E_1$, onde E_1 é um tipo entidade, E é um tipo entidade derivado, cuja população é o subconjunto da população de E_1 para o qual o

⁴ Produto.

⁵ Junção-relacional.

⁶ Junção identidade.

⁷ Estruturas não planares ou estruturas que não estão na primeira forma normal, são aquelas que permitem atributos compostos em sua estrutura.

predicado é verdadeiro. O esquema, os relacionamentos, generalizações e ligações de conjunção de E são derivados daqueles de E_1 .

O predicado pode envolver qualquer atributo ou atributo componente de E . Para cada atributo multivalorado uma variável associada com um quantificador (\exists ou \forall) deve ser definida.

A operação: $E = project$ [lista-atributos] E_1 , onde E_1 , é um tipo entidade e lista-atributos é uma lista de atributos de E_1 e/ou subestruturas de atributos de E_1 que criam um tipo entidade derivado E , cujos oids são os mesmos daqueles de E_1 . Os relacionamentos, generalizações e ligações de conjunção E são derivados daqueles de E_1 . O esquema de E é constituído dos atributos ou subatributos especificados na lista de atributos. As subestruturas da lista de atributos são definidas usando a notação convencional de ponto. O valor de cada ocorrência E é derivada dos valores das ocorrências correspondentes E_1 , restringindo conforme a lista de atributos.

Operador *reduction*⁹, é um operador novo. Este operador complementa as funcionalidades oferecida pelos operadores *selection* e *projection* com respeito a meta de selecionar a informação desejada de um tipo entidade. Através da seleção e projeção o usuário pode descartar ocorrências ou atributos que não lhe interessem, a redução lhe permite eliminar valores de atributos que não satisfazem um dado predicado.

A operação: $E = reduce$ [A / predicate] E_1 , onde E_1 é um tipo entidade e A é um atributo de E_1 , criado como um tipo derivado de E , cujos oids são os mesmos daqueles de E_1 . O esquema, relacionamentos, generalizações e ligações de conjunção E são derivados daqueles de E_1 . O predicado deve envolver A ou seus atributos componentes. Ele deve também envolver outros atributos de E_1 . Ele é definido como uma predicado de seleção. O valor de cada ocorrência de E é derivado dos valores das ocorrências correspondentes de E , mantendo em A somente os valores que satisfazem o predicado; o valor de outros atributos não é alterado.

⁸ Seleção.

⁹ Redução.

Operador *union*¹⁰, é diferente do operador da álgebra relacional, o qual é baseado em valor, pois mescla a população de duas entidades de acordo com seus oids. Ele é um operador binário com dois operandos principais. O ambiente do resultado (relacionamentos, generalizações e conjunções) é derivado de ambos os operandos.

A operação: $E = E_1 \text{ union } E_2$ onde E_1 e E_2 são dois tipos entidades, E é um tipo entidade derivado, cujos oids são os mesmos daqueles de E_1 acrescidos dos oids de E_2 . O esquema de E é derivado da união-fusão dos esquema E_1 e E_2 . Cada atributo A_i de E_1 (e de E_2) é também um atributo de E . Se E_2 não tem atributo com o mesmo nome, o atributo A_i de E é idêntico ao atributo A_i de E_1 (a única diferença é que A_i se torna opcional). Se E_2 também tem um atributo A_i , o atributo A_i de E é a união de ambos os atributos: seu domínio é a união de ambos os domínios, seu valor é a união multi-conjunto dos dois valores. Do mesmo modo, o relacionamento, generalização e ligações de conjunção E são derivados daqueles existentes em E_1 e E_2 pela união-fusão (fusão dos nomes idênticos e ligação dos mesmos tipos entidades).

Operador *r-join*¹¹, é um operador n-ário com um operando principal. Ele é usado para transformar uma rede de tipos entidades dentro de uma estrutura hierárquica (uma entidade simples). Assim este operador agrupa dentro de uma entidade simples a informação dispersa sobre as entidades ligadas através de um relacionamento. De um ponto de vista orientado a objeto, um *r-join* pode recompor como um simples objeto um objeto complexo que foi desmembrado dentro de componentes objeto.

A operação: $E = E_1 \text{ ü role}_1 \text{ r-join } (A : R, E_2 \text{ ü role}_2, \dots, E_n \text{ ü role}_n)$ onde role_i é o papel executado pelo tipo entidade E_i no tipo relacionamento (a especificação dos papéis é somente obrigatório em relacionamento cíclicos) e A é um nome para o novo atributo complexo que será criado, E é um tipo entidade derivado, cujos oids são os mesmo daqueles de E_1 (E_1 é o operando principal). Os relacionamentos, generalizações e ligações de conjunção de E são derivados dos pertencentes a E_1 . O esquema de E é constituído de atributos de E_1 mais um novo atributo

¹⁰ União.

¹¹ Join-relacional.

multivalorado denominado A , que é derivado dos esquemas de E_1, \dots, E_n e R . O valor de uma ocorrência de E é feita sobre os valores das ocorrências de E_1 mais um multiconjunto de E_1, \dots, E_n e ocorrências de valores de R que estão ligados a E_1 (se existir algum).

Operador *i-join*¹², é um operador binário com um operando principal. Ele é usado para agrupar dentro de uma entidade simples a informação dispersa sobre duas entidades delimitadas por uma ligação generalização ou conjunção. Permite ao usuário mesclar dois pontos de vista sobre um mesmo objeto do mundo real.

A operação: $E = E_1 \text{ i-join } (A : E_2)$, onde E_1 e E_2 são dois tipos entidades envolvidos por uma ligação *is-a* ou *may-be-a*, e A é um nome para o novo atributo complexo que será criado, E é um tipo entidade derivado, cujos oids são os mesmos dos de E_1 . Os relacionamentos, generalizações e ligações de conjunção de E são derivados daqueles de E_1 . O esquema de E é feito dos atributos de E_1 mais um novo atributo complexo monovalorado, denominado A , o qual é derivado dos esquemas de E_2 : ele agrupa todos os atributos de E_2 . O valor de uma ocorrência de E é feita sobre os valores das ocorrências correspondentes de E_1 adicionado do valor das ocorrências de E_2 (se existir).

Operador *product*, é usado para colocar tipos entidades não relacionados dentro de uma entidade simples. Este resultado é similar ao operador de produto relacional NF2¹³, porque cada entidade do primeiro operando está associado com todas as entidades do segundo operando. Ele é um operador binário com um operando principal (o primeiro). O produto é necessário para permitir ao usuário estabelecer dinamicamente ligações não previstas (não expressa por tipos relacionamento no esquema) entre tipos entidades não relacionados.

A operação: $E = E_1 (A : E_2)$, onde E_1 e E_2 são dois tipos entidade, e A é um nome para o novo atributo complexo que será criado, E é um tipo entidade derivado, cujos oids são os mesmos dos de E_1 (E_1 é o operando principal). Os relacionamentos, generalizações e ligações de conjunção de E são derivados daqueles de E_1 . O esquema de E é feito dos atributos de E_1 mais um novo atributo

¹² Junção-identidade.

¹³ Modelo relacional aninhado que descreve estruturas que não estão na primeira forma normal.

complexo multivalorado, denominado A , o qual é derivado do esquemas de E_2 . O valor de uma ocorrência de E é feita sobre os valores das ocorrências correspondentes de E_1 adicionado do valor das ocorrências do multiconjunto de todas as ocorrências de E_2 .

A álgebra ERC+ também contém dois operadores sintáticos, *renaming*¹⁴ e *simplification*¹⁵, os quais permitem 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 formular consultas mais curtas.

O operador *difference*¹⁶ (-) pode ser definido através da composição apropriada do produto, seleção e projeção. Uma *intersection*¹⁷ pode também ser definida 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 oids (interseção). Este dois operandos tem um operando principal, o primeiro.

O *r-join* é equivalente ao *outer join*¹⁸ do modelo relacional: o resultado tem uma ocorrência para cada ocorrência do operando principal mesmo que ele não esteja presente no relacionamento. Um operador derivado útil é *simplification* que exclui do resultado todas as ocorrências não ligadas através do relacionamento. O

¹⁴ Operador para troca de nome.

¹⁵ Simplificação.

¹⁶ Diferença.

¹⁷ Interseção.

¹⁸ Junção externa.

Do mesmo modo, outro operador derivado, *v-join*¹⁹, é uma junção teta NF2: ela agrupa dentro de um tipo entidade simples a informação dispersa sobre dois tipos entidades ligados por um predicado sobre seus atributos. É equivalente para uma expressão ERC+ feito sobre um produto seguido por uma redução e uma seleção.

O modelo 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 [].

Destaca-se na álgebra ERC+ o operador *r-join* pela sua importância em aplicações de consulta, pois a possibilidade de reduzir um conjunto de entidades, ligadas por tipos relacionamentos a uma única entidade permite uma grande flexibilidade, visto que se pode construir múltiplas visões hierárquicas sobre um mesmo esquema ERC+ exibido ao usuário, preservando a estrutura não planar obtida na resposta, mantendo portanto o conceito de objeto complexo do modelo.

¹⁸ Junção externa.

¹⁹ Junção por valor.

4 FERRAMENTAS DE CONSULTA A BANCO DE DADOS

Neste capítulo, as ferramentas de consulta a banco de dados são avaliadas e comparadas, no tocante à apresentação do resultado, a partir das características apontadas pelos seus autores, usando-se a abordagem da Engenharia Cognitiva (golfs de execução e de avaliação, distâncias semântica e articulatória) e da Engenharia Semiótica (conceitos gerais) e a partir da descrição dos protótipos [3],[16],[19].

Segundo de Souza [12], a avaliação da interface consiste em um importante passo do processo de *design*, pois é por meio dela que se consegue estimar o sucesso ou insucesso das hipóteses do *designer* sobre a solução que ele está propondo, tanto em termos de funcionalidade, quanto de interação. Ainda que o *designer* se baseie em uma abordagem teórica e conte com a ajuda de diretrizes e princípios de *design*, é necessário que ele avalie o resultado obtido

Os primeiros banco de dados baseados no CODASYL (*Conference on Data Systems Languages*) eram projetados por programadores [7]. Seus usuários eram seus próprios *designers*, que estavam preparados para entender as estruturas de dados armazenada, e capacitados a controlar exatamente qual nível de acesso pretendiam alcançar. Neste tempo (1969), a noção de interatividade não existia. Em seguida, o modelo relacional foi criado para beneficiar os não-programadores. Sua estrutura de dados é simples: ela é fácil de entender e oculta todos os itens de implementação; operações simples podem ser melhor expressas em declarações do que na forma procedural.

O modelo relacional foi o ponto de partida para o desenvolvimento de linguagens de consulta primeiramente formais (álgebra, cálculos) e depois orientadas ao usuário SQL (*Strutred Query Language*), QBE (*Query By Example*) e QUEL (*Query Language*). Estas linguagens tornam possível acessar dados no estilo pergunta-resposta, chegando àquelas mais usadas atualmente, de diagramas e ícones.

Sistemas comerciais de bancos de dados requerem uma linguagem de consulta mais "amigável ao usuário" sendo que as interfaces visuais surgiram após melhorias das soluções de hardware e software.

O paradigma da manipulação direta, onde o usuário movimenta com o *mouse* os objetos representados graficamente na tela do computador, rapidamente emergiu como um padrão de fato. A forma de representação não é uma questão fechada e tem sido objeto de pesquisa [19]. Os paradigmas mais utilizados para visualização das consultas funcionam por meio de formulários, ícones e diagramas, bem como pela combinação destes com abordagens em duas e três dimensões .

Muitos dos protótipos existentes são baseados em linguagens gráficas. Como consequência, a interação do usuário com o SGBD é construída a partir de uma representação diagramática do esquema do banco de dados. Quase todos os produtos usam diagramas ER para definição de dados e assumem a álgebra relacional para manipulá-los.

Aplicações tais como QBD*[3], SUPER[16] e VIQUEN[19] fazem uso da linguagem visual, com a qual o usuário interage para formular suas consultas, sem ser necessário conhecimento prévio da 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.

Algumas interfaces implementam mecanismos adicionais, tais como o modo de seleção automática (o sistema determina o melhor caminho de conexão para selecionar itens). QBD*[3] implementa uma técnica similar para selecionar *query* de sub-esquemas, bem como para a construção de expressões lógicas envolvendo entidades atributos (por exemplo predicados).

Para que os usuários interajam com as aplicações para bases de dados os conceitos do modelo de dados devem ser disponibilizados mediante representações, ou seja, a formulação de consultas e a apresentação dos resultados de forma visual exigem a representação do modelo escolhido.

A representação através de diagramas é a mais utilizada por sistemas de consulta visual [3],[16],[20] em virtude da facilidade de se relacionar os elementos do diagrama para expressar os diferentes conceitos do modelo e possibilitar a manipulação direta dos elementos do diagrama.

4.1 QBD*

QBD* (*Query by Diagram*) [3] é um sistema visual de consulta baseado na representação diagramática do esquema Entidade-relacionamento, descrevendo a estrutura interna do banco de dados. Permite ao usuário extrair informações de um banco de dados de forma amigável, mediante sua interface gráfica. Apresenta como características: a interface gráfica é uniforme tanto para a especificação do esquema quanto para formulação da consulta; utiliza um modelo de dados entidade-relacionamento orientado a objetos e permite consultas recursivas.

A facilidade de uso foi alcançada por meio de um ambiente totalmente gráfico. Neste ambiente o usuário interage com o sistema principalmente com o *mouse*, usando o teclado somente quando necessário. Para incrementar a parte “amigável” do sistema, um conjunto de facilidades de ajuda foi implementado. Dentre eles está a navegação *top-down* e a transformação do esquema.

A estrutura geral da consulta QBD* é baseada na localização de um conceito distinto chamado *main concept* (conceito principal, que pode ser uma entidade ou um relacionamento) que será visto como o ponto de entrada de uma ou mais sub-consultas; estas sub-consultas expressam possibilidades de navegação de um conceito principal para outros conceitos no esquema. Os atributos pertencentes a cada sub-consulta são determinados pela seguinte estratégia: os atributos do conceito principal são automaticamente considerados (a menos que explicitamente excluídos pelo usuário) enquanto os outros serão mostrados no resultado somente se requisitado pelo usuário.

A presença de um conceito principal associa um tipo com cada sub-consulta: como exemplo, se a entidade principal é a entidade “pessoa”, o resultado da consulta será um conjunto de pessoas (possivelmente enriquecido com atributos vindos de outros conceitos), não importando que tipo de operações subseqüentes o usuário executa para especificá-la.

Uma vez que o conceito principal tenha sido selecionado, dois diferentes tipos primitivos estarão disponíveis para navegação no esquema. O primeiro permite ao usuário seguir os caminhos existentes no esquema; o outro é usado para comparação dos dois conceitos para construção de um novo caminho (relacionamento) no esquema. Condições de comparação, bem como condições

genéricas dentro dos atributos, são expressos por meio de uma mecanismo simples de janelas incluindo um conjunto de ícones.

Uma vez realizada as operações de construção da consulta, os resultados desta, são mostrados na forma apresentada na FIGURA 4

CITY	STATE	DEPARTMENT	STUDENT	COURSE	TAKES
City	State	Department	Last Name	Name	Class Date
NEW YORK	NEW JERSEY	PHYSICS	ZARETSKY	CHEMISTRY I	35 CLASS
NEW YORK	NEW JERSEY	PHYSICS	SMITH	PHYSICS I	36 CLASS
NEW YORK	NEW JERSEY	PHYSICS	SMITH	ALGEBRA I	15 CLASS
NEW YORK	NEW JERSEY	PHYSICS	SMITH	GEOMETRY I	21 CLASS
NEW YORK	NEW JERSEY	PHYSICS	SMITH	PHYSICS II	23 CLASS
NEW YORK	NEW JERSEY	PHYSICS	SMITH	PROGRAMM	25 CLASS
NEW YORK	NEW JERSEY	PHYSICS	SMITH	ALGEBRA II	21 CLASS
NEW YORK	NEW JERSEY	PHYSICS	SMITH	NETWORKS	36 CLASS

FIGURA 4 - APRESENTAÇÃO DA RESPOSTA NO QBD* [8]

Tipicamente o grupo de aplicações de consulta para acesso a SGBDs relacionais, ao qual QBD* pertence, dedicam-se somente à formulação de consultas, desconsiderando a exibição de respostas

Essas aplicações apresentam respostas planares para que o usuário as interprete ou formate sua exibição em ferramentas de relatórios, exigindo conhecimentos razoáveis de ordenação por múltiplos atributos, quebras em relatórios, etc. Além disto a ferramenta QBD*, emprega o modelo entidade-relacionamento como modelo conceitual, o qual não suporta objetos complexos em sua semântica, característica que o torna deficitário em relação a aplicações de consulta que se valem de modelos orientados a objetos/objeto-relacionais.

A ferramenta QBD* exhibe o resultado das consultas de forma bidimensional. Esta forma de representação de consulta visual é baseada na exibição do esquema de relações, com linhas e colunas necessárias para representar as tuplas (registros) a serem definidos.

Como mostrado na FIGURA 4 este formato de exibição gera dificuldades de interpretação para o usuário. Esta dificuldade é causada pelo fato de os conteúdos referentes a um campo aparecem repetidamente até que as ocorrências tenham terminado.

Mesmo apoiado nas facilidades de ajuda disponíveis na ferramenta, este formato não se mostra eficiente para o usuário, pois este necessita “filtrar” as informações apresentadas para avaliar se a saída está compatível com o resultado esperado por ele.

4.2 SUPER

SUPER [16] pertence à família dos editores de consultas. É um editor visual, onde o termo “visual” é utilizado para enfatizar todos os aspectos da interação, e não somente a representação diagramática do esquema do banco de dados. Por exemplo, o uso de menus e botões foi minimizado, ficando limitado ao gerenciamento de comandos de arquivos (salvar, abrir e outros). A manipulação direta de objetos na área de trabalho é a principal técnica de interação para o usuário que o usuário tem à sua disposição.

A interface de consulta é consistente com a definição de dados da interface, ou seja, ela interage com os usuários baseando-se em um conjunto comum de conceitos de modelagem de dados utilizando o modelo ERC+ [37] para interagir com usuário e não somente para exibir o esquema completo da base. A formulação de consulta foi projetada para que o usuário não necessite conhecer a linguagem subjacente de manipulação de dados possuindo alta flexibilidade na implementação das tarefas, permitindo que este possa desfazer (*undo*) parte de sua interação.

A especificação e execução de uma consulta incluem as seguintes ações: seleção do sub-esquema para a consulta, transformação do sub-esquema em uma estrutura hierárquica, especificação de predicados, definição do tipo de resultado e apresentação do resultado.

O usuário, além de poder ver todos os elementos relacionados a um elemento corrente, pode manipular suas visões de estrutura de atributos complexos multi-valorados, podendo também optar por diferentes visões do objeto, alternando o tipo de acordo com a hierarquia de generalização.

O navegador SUPER implementa dois modos de visualização de dados. No modo baseado em formulário, ocorrências são apresentadas com formas de apresentação do objeto correspondente ou tipo de relacionamento, enquanto no modo gráfico eles são apresentados como diagramas entidade-relacionamento mostrando as ocorrências que estão sendo examinadas.

No modo baseado em formulários, a ocorrência ativa é mostrada em um objeto formulário. Consiste de uma área de comandos, uma área de atributos, uma área de regras, e se o ponto de entrada é um tipo objeto, uma área de multi-instâncias.

A área de comandos inclui uma barra de rolagem horizontal para navegar pelo conjunto de ocorrências, o número de ocorrências da população ativa, o número relativo à ocorrência ativa e os botões de classificação e filtro da população ativa.

Na área de atributos os nomes dos atributos de primeiro nível são listados e os valores de atributos simples e mono-valorados são exibidos; atributos complexos e multi-valorados estão ocultos. O usuário pode exibí-los "clitando" no botão associado; a estrutura de atributos complexos é então exibida e uma barra de rolagem horizontal é associada aos atributos multi-valorados podendo-se verificar seus valores.

Na área de regras estão os nomes das regras do tipo objeto listado. Eles estão inicialmente ocultos e podem ser exibidos de duas maneiras diferentes. Com um "clique" na regra o usuário pode ver as instancias ligadas à ocorrência ativa sem abrir um novo formulário; isto é, sem mudar a população ativa. Estas instâncias são exibidas na área de regras.

Se o usuário faz um "clique" duplo na regra, o conjunto de ocorrências ligadas à ocorrência ativa se torna a população ativa, sua primeira instância se torna a ocorrência ativa e ela é mostrada em um novo formulário. A área de multi-instâncias é definida somente por formulários e listas de nomes de objetos ligados através de generalização/especialização ou ligações de multi-instanciação.

A resposta no modo formulário é mostrada na FIGURA 5.

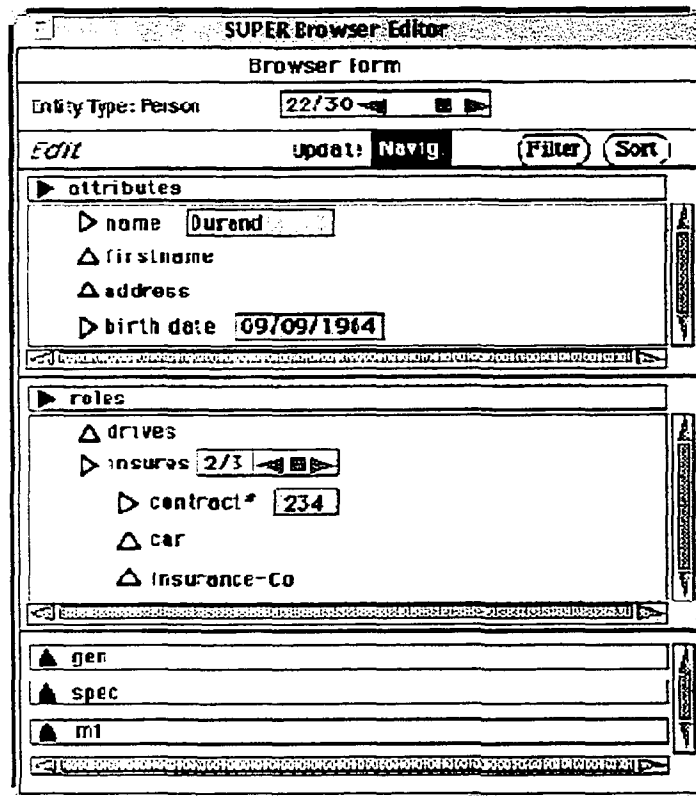


FIGURA 5 - RESPOSTA NO MODO FORMULÁRIO – SUPER [16]

No modo gráfico as funções são as mesmas que no modo baseado em formulário, mas realizadas com uma abordagem de manipulação direta similar ao diagrama ERC+. O ponto de entrada é exibido em uma janela com símbolos gráficos derivados da construção diagramática do tipo correspondente e é representado inicialmente como atributos e valores, quatro botões para navegação da população e dois números indicando o número de instâncias da população ativa e a posição relativa da instância ativa.

Por exemplo, para mostrar a instância “pessoa” o usuário clica no botão “>” (próximo) do tipo objeto “pessoa”. A próxima instância se torna a instância ativa e pela facilidade de sincronização, a atualização é propagada para o relacionamento “seguro”, e este, exhibe o objeto “carro” e “companhia-de-seguro”. As regras e ligações de multi-instanciação são mostradas somente após o usuário ter especificado que ele que navegar através do banco de dados. Estas ligações são sensíveis ao *mouse*, assim, quando o usuário aponta, ele indica qual caminho quer seguir.

Como o navegador é usado para manipular dados, no modo de navegação a estrutura de um atributo é exibida somente se valores existirem na instância corrente. Do mesmo modo, regras e ligações de multi-instâncias são mostradas somente se a ocorrência ativa é ligada a outras ocorrências através destas ligações.

A resposta no modo gráfico é mostrada na FIGURA 6.

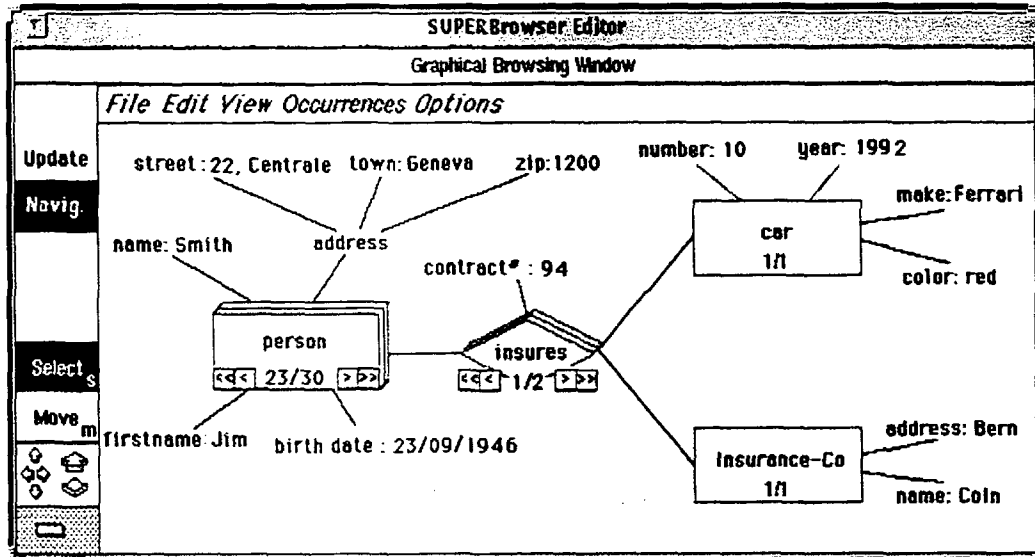


FIGURA 6 - RESPOSTA NO MODO GRÁFICO – SUPER [16]

A ferramenta SUPER implementa dois modos de visualização de dados.

No modo baseado em formulário, o usuário tem maior dificuldade de interpretação do resultado da consulta pois quando os atributos são complexos e multi-valorados estes necessitam ser expandidos para mostrar os dados ocultos.

Além de observar que existem dados ocultos, o usuário deve também conhecer conceitos específicos de Banco de Dados, tais como "generalização" e "especialização", utilizados durante a interação.

No modo gráfico a saída é apresentada como diagramas entidade-relacionamento mostrando as ocorrências que estão sendo examinadas, realizadas com uma abordagem de manipulação direta similar ao tratamento no diagrama ERC+. A navegação no nível de cada entidade ou atributo dificulta a visualização de conjunto de ocorrências da resposta

4.3 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 [18] com o objetivo de atingir a categoria de usuários comuns, os quais buscam freqüentemente 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, tais como Pasta-3 [20], QBD*[3] e SUPER[16], que fazem uso da linguagem visual com a qual o usuário interage para formular suas consultas, sem ser necessário conhecimento prévio da 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 através 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+[37].

O modelo ERC+ foi utilizado em [19] no desenvolvimento do VIQUEN como apoio à melhoria da interação usuário-sistema, ou seja, é por meio dos conceitos deste modelo que o usuário formulará suas consultas e interpretará as respostas.

O protótipo VIQUEN apresenta as seguintes etapas na interface com o usuário: 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 e a visualização da resposta.

No processo de engenharia reversa, o esquema contendo todas as tabelas e restrições pertencentes ao esquema selecionado pelo usuário são traduzidas automaticamente pela aplicação para um esquema sob o modelo ERC+. Nesta representação podem ser visualizados entidades, relacionamentos e atributos complexos que são apresentados graficamente ao usuário em uma janela. As ligações entre os elementos desta janela são feitas por meio de arestas. Em uma outra janela, ativada pelo usuário por meio de uma caixa de diálogo, 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.

A partir do esquema recuperado, o usuário pode selecionar, mediante o *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. O usuário também pode selecionar os atributos relevantes de cada objeto que devem participar da consulta. Para submeter uma consulta o usuário deve determinar, também, qual a visão desejada, escolhendo uma raiz para a consulta. A seleção da raiz da consulta constitui uma etapa importante na seleção do esquema, pois é esta escolha que irá determinar a hierarquia dos objetos participantes da consulta a ser realizada. Em alguns casos, é preciso que o usuário elimine eventuais ciclos que ocorram entre os objetos da consulta. Por último, a fim de complementar a consulta, o usuário pode adicionar sub-esquemas, que devem ser unidos ao sub-esquema inicial, eliminando objetos comuns a ambos.

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 por meio da seleção dos operadores lógicos e relacionais e de constantes, para construir expressões *booleanas* sobre os atributos atômicos. Feito isto, são 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 construção SQL que foi automaticamente gerada pela aplicação.

A última etapa é a visualização da resposta, onde os objetos selecionados durante a consulta permanecem igualmente vistos como objetos. Esta representação em estrutura não planar, mas sim de forma hierárquica, a partir da raiz previamente selecionada, possibilita que o usuário expanda as instâncias da resposta, visualizando todas as ocorrências que satisfazem ao predicado por ele especificado. A apresentação da resposta no VIQUEN [19], está mostrada na FIGURA 7.

Objetos	Valores
[-] PESQUISADOR	
[-] PRIMEIRO_NOME	Maria
[-] SEGUNDO_NOME	José
[-] SOBRENOME	Flores
[-] LINHAPESQ	Exibição de Dados
[-] REALIZA	
[-] TRABALHO	
[-] TITULO	Uma Interface para Exibição
[-] ANO	2001
[-] ARTIGO	
[-] LOCAL	São Paulo
[-] TRABALHO	
[-] TITULO	Bases Multidimensionais
[-] ANO	2001
[-] ARTIGO	
[-] LOCAL	Fortaleza

Objetos	Valores
[-] PESQUISADOR	
[-] PRIMEIRO_NOME	João
[-] SOBRENOME	Lara
[-] PERTENCE	
[-] INST_ENSINO	
[-] NOME	UNIOESTE
[-] ENDER	Cascavel
[-] REALIZA	
[-] TRABALHO	
[-] TITULO	Threads: Um Release
[-] ANO	2001
[-] PESQUISADOR	
[-] PRIMEIRO_NOME	André
[-] SOBRENOME	Ribeiro
[-] PERTENCE	
[-] INST_ENSINO	
[-] REALIZA	
[-] TRABALHO	
[-] TITULO	Processadores Múltiplos
[-] ANO	2001

FIGURA 7 - APRESENTAÇÃO DA RESPOSTA NO VIQUEN [19]

Na etapa de visualização da resposta, o usuário pode avaliar seus valores, expandindo suas ocorrências. A interface não revela ao usuário que ele tem essa ação disponível. A técnica de expansão é simples porém, caso o número de ocorrências seja elevado, a visualização simultânea de todas as ocorrências torna-se dificultada pela estrutura de resposta apresentada, aumentando a complexidade da avaliação do resultado

A técnica de visualização das respostas pela expansão não permite que se observe um número razoável de ocorrências, limitando-o ao espaço disponível em tela. Esta técnica poderia ser implementada ou pela inclusão de novos *widgets* presentes nas linguagens visuais de desenvolvimento que possibilitem expressar a meta-comunicação *designer-usuário* necessária no ambiente e revelem as possibilidades de tratamento das respostas. Facilidades de ajuda não estão disponíveis na exibição do resultado (como as apresentadas na ferramenta QBD*).

A seqüência LEMD para a etapa da avaliação do resultado da consulta é descrita a seguir, mostrando que a interface apresenta problemas de expressão na mensagem do *designer*-usuário. Os problemas encontrados estão destacados pelo sublinhado duplo.

A mensagem passada pelo designer ao usuário é insuficiente para permitir a interpretação da resposta, pois lhe é proporcionado somente um modo de visualização das ocorrências. Além disto, pode não observar que existem mais ocorrências a serem visualizadas por meio da expansão das mesmas pelo uso do sinal "+".

// Início Resultado

Command-Message

{

View Hierarchy-of <Estrutura do Resultado>

Activate Stop Application-function Consulta ERC+

}

// Término Resultado

5 A INTERFACE PARA A APRESENTAÇÃO DO RESULTADO

A análise das ferramentas para consulta, identificou problemas de interação, que não impedem o usuário de explorar a potencialidade das ferramentas mas que aumentam a carga cognitiva deste na direção da mesma.

A opção por dispensar o usuário do aprendizado de sintaxes para execução de tarefas, presentes nos ambientes convencionais, tais como a substituição de convenções sintáticas formais por operações intuitivas e a atividade de memorizar a estrutura do banco de dados, facilita a interpretação do resultado da consulta.

Este trabalho pretende manter esta opção e contribuir com uma interface que disponibilize ao usuário três formas para visualização do resultado da consulta. A idéia é proporcionar flexibilidade e diminuição do esforço cognitivo do usuário pela utilização da forma que mais se adapte ao seu perfil e ao seu conhecimento dos termos relativos a Banco de Dados. Estas três formas diferenciam-se apenas na maneira pela qual os dados são apresentados, ou seja, o conteúdo da resposta será sempre o mesmo independentemente da forma de visualização escolhida pelo usuário.

5.1 DOMÍNIO DA APLICAÇÃO

O domínio tomado como hipótese para a construção do protótipo foi o de um sistema de controle de publicações da associação de professores universitários do estado do Paraná. O modelo [35] foi adaptado e submetido a ferramenta ERC+[19], para geração da resposta à consulta e está representado na FIGURA 8:

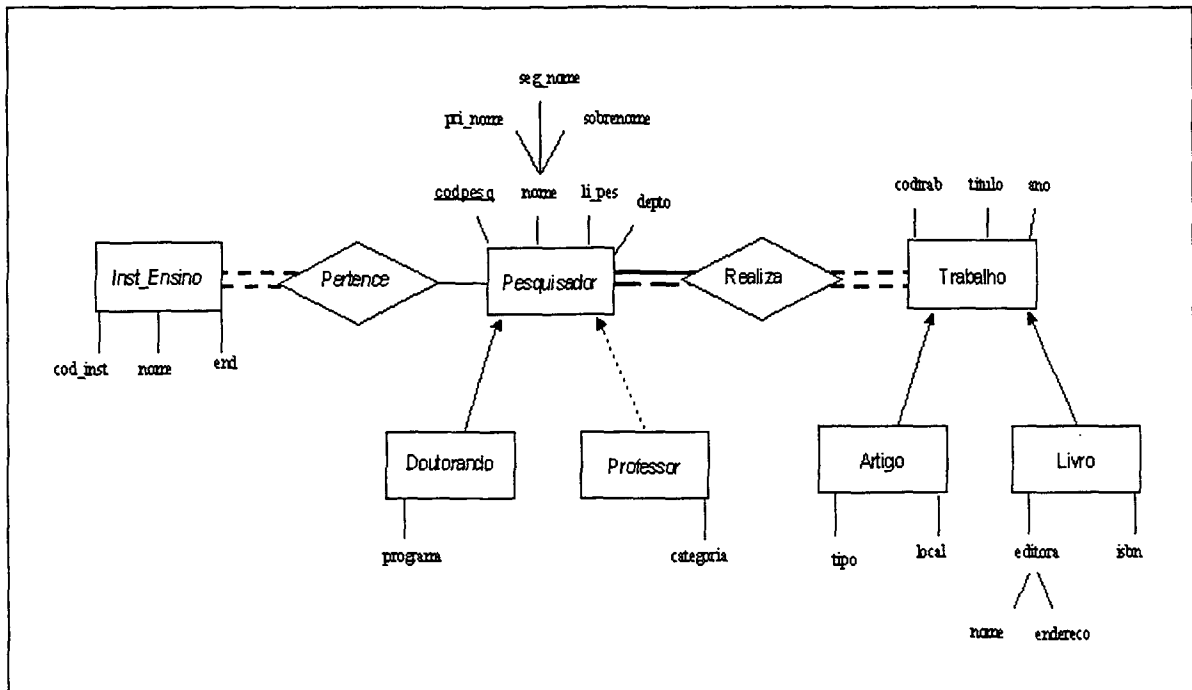


FIGURA 8 - DIAGRAMA ERC+ DO DOMÍNIO DA APLICAÇÃO [35]

Na FIGURA 8 são observados conceitos do modelo ERC+, como a entidade Pesquisador ligada à entidade Trabalho pelo relacionamento Realiza. Este relacionamento pode ser expresso pelas seguintes semânticas: “um pesquisador pode realizar zero ou mais trabalhos”, e ainda, “um trabalho pode ser realizado por um ou mais pesquisadores”.

A entidade Pesquisador é uma generalização das entidades Doutorando e Professor. A seta formada por uma linha contínua indica uma generalização *is-a*, expressando que todos os doutorandos são pesquisadores. A seta formada por uma linha tracejada indica uma generalização *may-be-a* expressando que os professores podem ou não ser pesquisadores.

Ainda, na entidade Pesquisador, podemos verificar a existência de um atributo complexo, **nome**. Este atributo possui dois atributos obrigatórios, **pri_nome** e **sobrenome**, e um atributo opcional **seg_nome**.

A entidade Trabalho é uma generalização das entidades Artigo e Livro. A ligação entre elas é uma seta formada por uma linha contínua indicando uma generalização *is-a*, onde todos os artigos e livros são considerados trabalhos.

Os atributos codpesq e codtrab são atributos chaves das entidades Pesquisador e Trabalho.

5.2 PERFIL DO USUÁRIO

Quando uma interface usuário é projetada, é necessário levar-se em consideração o perfil dos seus usuários potenciais. A análise do perfil do usuário é crucial para que a interface e, conseqüentemente, o software como um todo, sejam explorados em todo o seu potencial.

Assim, "todo projeto deve iniciar-se com um entendimento do perfil dos usuários a que se destina (idade, sexo, capacidades físicas, educação, *background* cultural ou étnico, motivação, metas e personalidade) entre outros" [36]. Do ponto de vista da familiaridade com o sistema em questão, os usuários podem ser classificados, como segue.

Usuários novatos ou principiantes

Os usuários novatos ou principiantes possuem pouco conhecimento sintático²⁰ e semântico²¹ da aplicação ou uso do computador em geral. Ele necessita de facilidades para fazer bom uso da interface entre as quais destacam-se: necessidade de simplificação, abstração de dificuldades, auxílio *on-line*.

Usuários intermitentes

Estes usuários possuem razoável conhecimento semântico da aplicação, mas relativamente pouca lembrança de informações sintáticas para usar a interface. Para o usuário intermitente, é necessário que a interface ofereça: consistência, mensagens informativas (*feedback*), manuais de referência para recuperação da sintaxe.

Usuários experientes

Os usuários freqüentes possuem bom conhecimento semântico e sintático, procurando freqüentemente por atalhos e modos abreviados de interação. Para estes usuários as características seguintes tornam o uso da interface mais produtivo: enunciado das tarefas facilitado, atalhos, eficiência na resposta.

²⁰ Neste contexto, conhecimento sintático refere-se às regras de interação exigidas para usar a interface eficientemente.

²¹ Conhecimento semântico refere-se à percepção subjacente à aplicação - uma compreensão das funções que são executadas, do significado de entrada e saída e das metas e objetivos do sistema.

Na construção deste protótipo, considerou-se que o usuário possui 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 o protótipo foi desenvolvido em uma ferramenta voltada para ambiente Windows²², considerou-se também a construção de uma interface gráfica: *GUI (Graphical User Interface)* no estilo *WIMP*. Além disto a aplicação é destinada a computadores pessoais, com usuário único, que possua conhecimento médio do estilo de interação proposto na interface.

5.3 ANÁLISE DE TAREFAS

O objetivo da análise de tarefas é fornecer ao *designer* a visão das tarefas que os usuários precisam realizar [13]. Baseado no domínio escolhido para o presente trabalho, para a apresentação da resposta da consulta à base de dados no modelo ERC+, foi identificada uma grande tarefa (visualização da resposta) e foram definidas as seguintes ações a serem disponibilizadas ao usuário no QUADRO 1:

VISUALIZAÇÃO DA RESPOSTA
<ol style="list-style-type: none"> 1. Indicar modo de visualização da resposta; 2. Efetivar a escolha; 3. Voltar (Cancelar) à formação do predicado; 4. Solicitar Auxílio (Ajuda) sobre a Etapa; 5. Navegar pelas ocorrências; 6. Alternar modo de visualização; 7. Selecionar Modos de Visualização; 8. Elaborar Novo Predicado; 9. Solicitar Auxílio sobre a Etapa; 10. Elaborar Nova Consulta.

QUADRO 1 – AÇÕES DISPONÍVEIS AO USUÁRIO NA VISUALIZAÇÃO DA RESPOSTA

²² Este ambiente operacional é atualmente o mais difundido em computadores de pequeno porte.

5.4 LEMD DA INTERFACE PARA VISUALIZAÇÃO DA RESPOSTA

A LEMD funciona como sistema semântico que permite descrever estes componentes do objeto da mensagem. Isto significa que a cada elemento do modelo de usabilidade que tenha sido concebido e especificado utilizando os tipos semânticos da LEMD devem estar associados por regras de correlação semântica a um signo de interface. Signos do domínio, funções aplicadas, visualizações e, principalmente, comandos de funções devem poder ser descritos na LEMD para serem representados por *widgets*. A LEMD completa da interface para apresentação do resultado está apresentada no Apêndice 2 do presente trabalho.

5.5 REGRAS DE CORRELAÇÃO (MAPEAMENTO SEMÂNTICO)

Como pode ser observado, a representação do conteúdo da mensagem, é feita de maneira independente de sua expressão, isto é, independente de como será materializada no *medium* computacional. Para que o *designer* possa associar os termos utilizados na LEMD, com um elemento de interface (um *widget*, no caso do padrão Windows) ele necessita utilizar regras de correlação semântica. Estas regras visam descrever os significados esperados para os principais signos utilizados no desenvolvimento do protótipo e são descritas no QUADRO 2 :

Elemento da LEMD	<i>Widget</i> associado	Significado
<i>View</i>	<i>Labels</i> ou <i>message-boxes</i>	Mensagem direta do <i>designer</i> para o usuário
<i>Activate</i>	Botões de pressão	Acionamento de uma função da aplicação
<i>Select information-of</i>	<i>Combo-boxes</i> ou painéis	Escolha de uma opção de uma lista
<i>Select set-of</i>	<i>Check-boxes</i>	Escolha através da marcação de uma opção
<i>Join</i>	Painéis ou <i>boxes</i>	Agrupamento
<i>Sequence</i>	Painéis ou <i>boxes</i>	Seqüência obrigatória de execução de tarefas.
<i>Combine</i>	Painéis ou <i>boxes</i>	<i>Layout</i> de agrupamento de <i>widgets</i>

QUADRO 2 - REGRAS DE CORRELAÇÃO SEMÂNTICA

5.6 STAG DA INTERFACE PARA VISUALIZAÇÃO DA RESPOSTA

A partir do uso da ferramenta STAG é possível aliar a descrição da LV à gramática necessária para a proposta da interface, objeto deste estudo.

É importante salientar, conforme Martins em [22], que a STAG é uma notação de auxílio, não se propondo a determinar o que deve ser usado. Mesmo sendo implementada através de um aplicativo, ela não prescreve soluções para o *designer*. Ela apenas sugere soluções mais adequadas de acordo com suas heurísticas, porém, a decisão final a respeito da LV, será sempre do *designer*. A STAG completa da interface para apresentação do resultado está apresentada no Apêndice 3 do presente trabalho.

5.7 A INTERFACE PROPOSTA

5.7.1 Caracterização geral

Conforme descrito em [32], a confecção de auxílios e manuais também pode ser considerado parte do processo de *design* de interfaces. Auxílios e manuais podem derivar tanto do designer, quanto do modelo de usabilidade por ele estruturado e até mesmo da interface já desenvolvida. A FIGURA 9 mostra como este processo é estruturado.

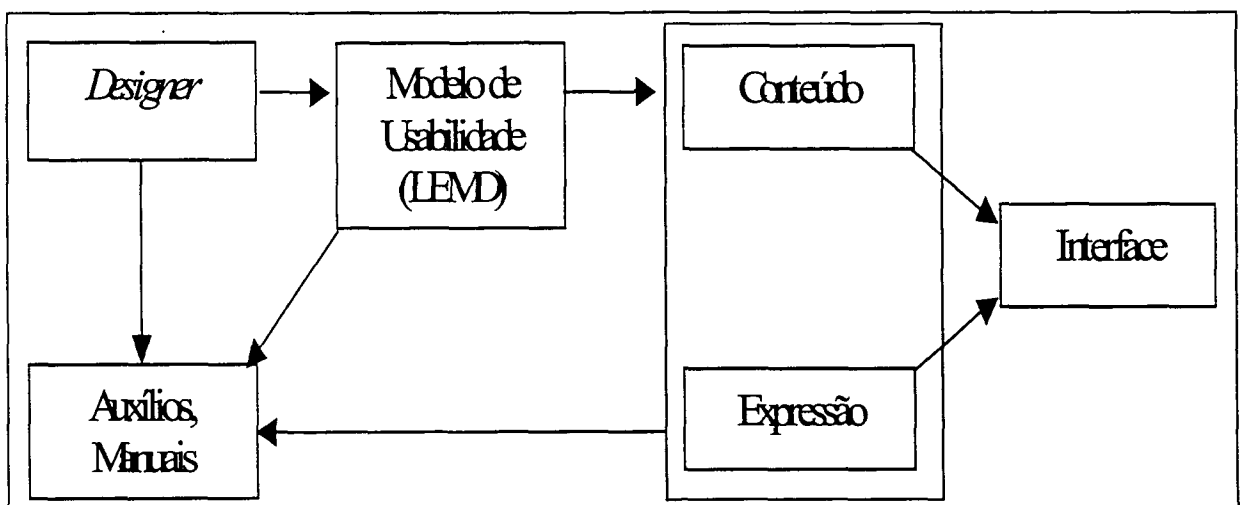






FIGURA 9 – PROCESSO DE DESIGN DA INTERFACE [32]

Após o usuário ter executado as etapas anteriores (seleção do sub-esquema [6] e formação do predicado [7]), este poderá visualizar o resultado da consulta.

Inicialmente, será apresentada a janela na qual o usuário poderá optar pela forma com que ele deseja que sejam apresentados os resultados. Estas formas, denominadas Modo Formulário, Modo Gráfico e Modo Painéis respectivamente serão detalhadas mais adiante.

A opção por um dos modos de visualização é feita em uma caixa de grupo contendo um conjunto de caixas de verificação (*check-box*). Esta seleção será exclusiva, ou seja, nunca haverá dois modos selecionados simultaneamente. Entretanto esta seleção poderá ser refeita pelo usuário a qualquer momento.

Uma vez feita a opção, mas ainda antes de efetiva-la a maneira como serão visualizados os dados será mostrada ao usuário por meio de um exemplo (visão prévia). Abaixo do grupo “Modos Disponíveis” um botão com o nome do modo escolhido será ativado. Um clique neste botão, permite ao usuário efetivar sua escolha, levando-o então à apresentação do resultado.

Há ainda na janela um painel contendo um grupo de botões denominado “Ações”. Neste grupo, o botão  permite a volta a ação de seleção do modo de exibição. O botão  possibilita ao usuário abandonar a etapa de Visualização da Resposta, retornando à etapa de elaboração do predicado. O botão  fornece auxílio ao usuário, indicando procedimentos para execução da etapa. Por fim o botão  possibilita ao usuário abandonar a visualização da resposta, voltando primeiro passo para a elaboração de uma consulta.

A janela apresenta também uma barra de estado do sistema (*status bar*) que mostra, além da data e hora correntes, descrições funcionais dos componentes de interação (*widgets*) presentes na janela. A janela inicial do protótipo é apresentada na FIGURA 10.

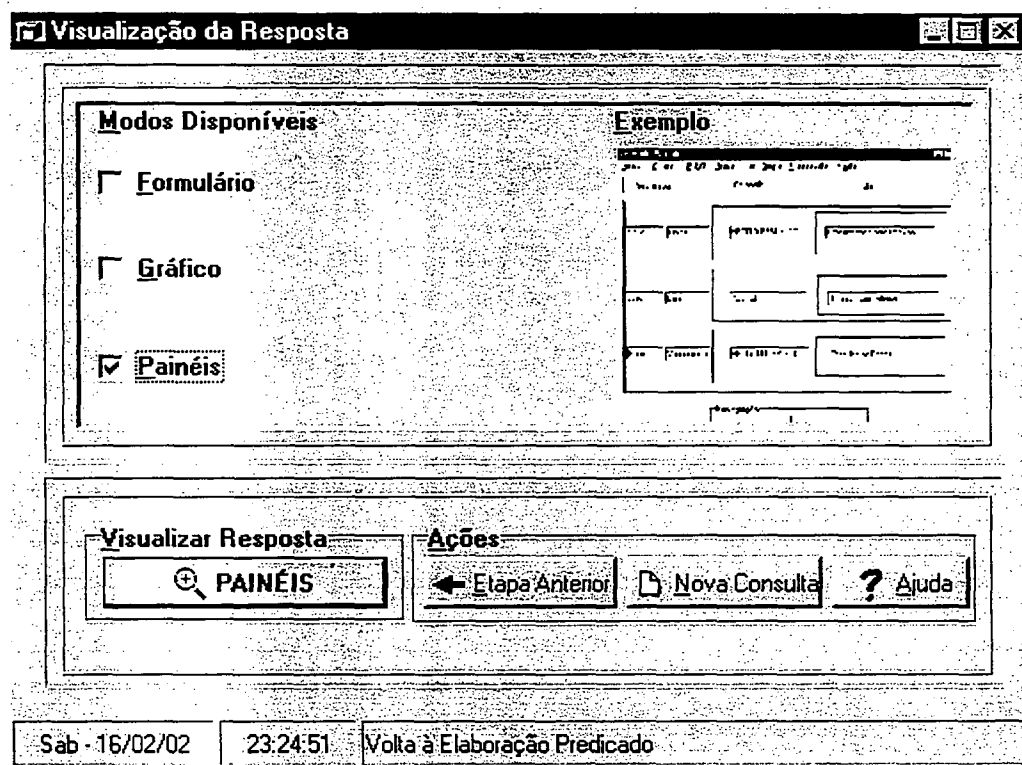





FIGURA 10 – JANELA INICIAL DA RESPOSTA

As janelas dos três modos de visualização (Formulário, Gráfico e Painéis) apresentam pontos em comum, objetivando diminuir o esforço cognitivo do usuário quando este opta por um ou outro modo. Estas características são descritas a seguir.

Uma barra de títulos indicando a etapa em execução no momento corrente (Visualização da Resposta). Uma barra de menus com as opções: **Arquivo**, **Sub-Eschema**, **Predicado**, **Resposta** e **Ajuda** onde o usuário tem disponíveis as mesmas funções presentes na interface quando interage através da manipulação dos objetos e botões presentes na interface.

Um painel contendo dois grupos de botões: o primeiro, denominado “Modos de Visualização” onde o usuário pode alternar entre os modos de visualização da resposta migrando entre os três disponíveis, permitindo a este ter acesso a todas as visões a qualquer momento da interação, garantindo-lhe maior flexibilidade durante a análise. No segundo, denominado “Ações”, o usuário tem à sua disposição as ações disponíveis durante a interação.

Neste segundo grupo, o botão  fornece auxílio ao usuário, indicando procedimentos para execução da etapa. O auxílio também estará disponível através de *hint*, apresentando o significado ou função de cada componente presente na interface seja ele de interação ou relativo ao modelo ERC+ na área destinada à exibição de mensagens.


Visando tornar a aplicação mais eficiente esta foi dotada da capacidade de desfazer uma ação ativada por engano. Este recurso garante ao usuário que, na ocorrência de erro, ele possa retornar à etapa anterior. A opção de desfazer (*undo*) é representada no ambiente pelo botão . A volta atrás pode ser efetivada no nível de tarefas por meio da pressão do botão .

Todas as telas da aplicação foram dotadas de uma barra de títulos padrão informando o modo de visualização corrente. Além disto, em termos de orientação ao usuário, todas as janelas dispõem de uma barra de estado do sistema (*status bar*) onde são mostradas as mensagens de alerta e execução no momento da interação. Outro facilitador é o fato de, em cada janela, estar habilitado somente o botão que pode exercer ação na janela. Os botões não habilitados estão com sua cor alterada para cinza claro.

Para os usuários mais familiarizados (experientes) com a hierarquia de menus, as opções de interação também estão presentes. Nos menus, estão habilitadas somente opções disponíveis naquele momento e as não disponíveis estão com sua cor também alterada para o cinza claro, a exemplo dos botões. Os acessos às opções de menu, estão disponíveis também através de atalhos ou teclas aceleradoras (*hot keys*). A combinação da tecla ALT com a letra sublinhada no menu, ativa este menu para o usuário. Para o usuário acessar a Ajuda ele pode clicar diretamente na opção de menu ou utilizar a tecla aceleradora formada pela pressão das teclas ALT e U.

As telas dos três modos de visualização apresentam dois botões que permitem ao usuário visualizar as ocorrências que satisfazem ao predicado estabelecido pelo usuário e também uma caixa denominada **No. de Ocorrências**

(atual e total) onde o usuário verifica em qual ocorrência ele está no momento e quantas ocorrências satisfazem ao predicado formulado.

A aplicação disponibiliza um assistente, presente em cada tela, ativado de duas maneiras: pela pressão do botão  e por meio do menu Ajuda. Com este recurso todas as ações a serem realizadas para executar determinada tarefa, bem como informações genéricas sobre a aplicação, são disponibilizados ao usuário.

Outro recurso de auxílio fornecido pela aplicação é o denominado *hint* (dica). Este recurso mostra ao usuário uma breve mensagem sobre o *widget* no qual o cursor está posicionado bastando, para isso, que o usuário permaneça com o cursor sobre o *widget* por algum tempo.

O uso do auxílio sensível ao contexto por meio de caixas de diálogo permite ao usuário acionar a ajuda quando achar conveniente, possibilitando ao usuário conhecer as ações disponíveis no instante de interação corrente.

Na FIGURA 11 é mostrado um exemplo da interação com o auxílio do ambiente. Este é constituído de *links* (sublinhados e na cor verde) onde o usuário clica para obter informações sobre o tópico. O auxílio completo é apresentado no Apêndice 1 do presente trabalho.

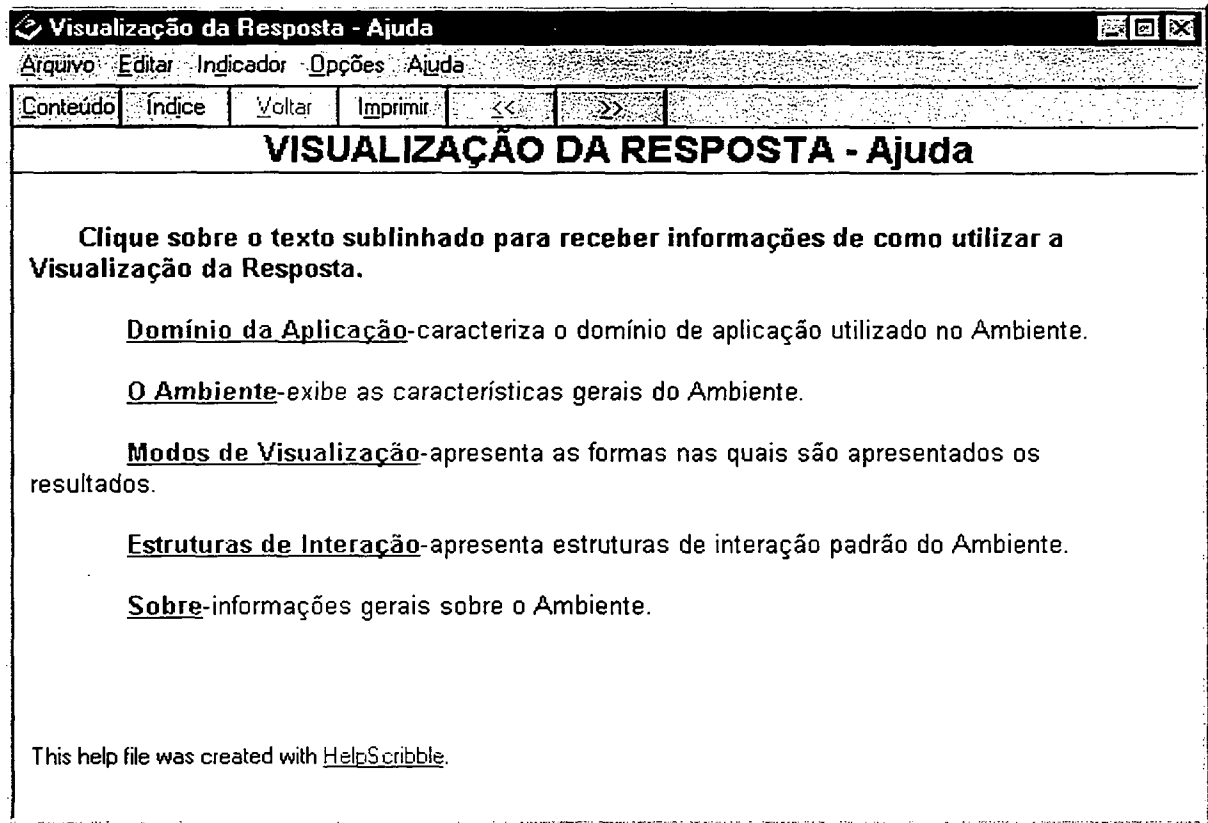


FIGURA 11 – JANELA PRINCIPAL DO HELP

5.7.2 Modos de exibição

Para ilustrar as respostas nos três modos de visualização, formulou-se uma consulta hipotética à base de dados com os seguintes requisitos:

“Apresente o nome e sobrenome do Pesquisador, título e ano do Trabalho, tipo e local do Artigo dos trabalhos publicados nos anos de 2000 e 2001” .

A seguir são apresentados os modos de visualização e as considerações sobre a interface de cada um deles.

Modo Formulário

O modo de visualização Formulário apresenta ao usuário as ocorrências por meio de grades (*grids*). A entidade principal escolhida na etapa da seleção do sub-esquema (raiz da consulta) é a primeira a aparecer no formulário.

A primeira ocorrência que satisfaz o predicado formulado fica em destaque, utilizando-se para isto dois recursos: o realce da ocorrência com cor diferenciada (azul) e o tamanho da sua fonte.

As ocorrências relativas às demais entidades que compõem o sub-esquema selecionado também são mostradas em grade, respeitando o predicado definido pelo usuário. Para a visualização os objetos complexos, utilizam-se as barras de rolagem que serão apresentadas nas grades que conterão tais atributos.

Os rótulos das grades são gerados a partir dos objetos escolhidos pelo usuário durante a seleção do sub-esquema. O objetivo é manter a consistência diminuindo assim o esforço cognitivo do usuário durante a interação, visto que, este modo de apresentação da resposta, não apresenta os elementos visuais que caracterizam as entidades no modelo ERC+[37]. A janela do modo formulário está apresentada na FIGURA 12.

Este modo de visualização estrutura a resposta, se constituindo em uma solução de mais fácil interpretação do que uma resposta em forma planar, tradicional. Ela atende aos usuários experientes, pois requer memorização dos tipos (entidade, atributo) dos elementos selecionados no sub-esquema e exige visão de conjunto da resposta ampliada. Este modo privilegia, também, a visualização de respostas com muitas ocorrências, sem focalizar os aspectos intrínsecos do modelo ERC+.

Visualização da Resposta - Modo Formulário

Arquivo Sub-Esquema Predicado Resposta Ajuda

Pesquisador

Nome	Sobrenome
▶ Lucas	Ranthum

Trabalho

Título	Ano
▶ Uma Interface para B.D.	2000
Paradigmas E. Software	2001

Artigo

Tipo	Local
▶ Poster	UFPR
Art. Resumido	UEPG

Navegação

◀ Anterior ▶ Próximo

No. de Ocorrências

1 / 5

Modos de Visualização

Gráfico Painéis

Ações

✕ Cancelar ◀ Etapa Anterior ▶ Nova Consulta ? Ajuda

Sab - 16/02/02 23:25:45 Alternar para Visualização no Modo Gráfico

FIGURA 12 – JANELA DA RESPOSTA NO MODO FORMULÁRIO

Modo Gráfico

O modo de visualização Gráfico permite ao usuário ter as ocorrências dispostas em torno dos elementos do sub-esquema com uma abordagem de manipulação direta similar ao da manipulação dos objetos no diagrama ERC+ [37]. O sub-esquema selecionado será transportado com os símbolos gráficos derivados da construção diagramática para a janela da resposta, apresentando-os de maneira similar ao protótipo construído para apresentação do ERC+. O recurso gráfico permite que a consistência seja mantida durante todas as fases da interação, permitindo que o usuário associe os elementos do sub-esquema com as ocorrências mostradas que satisfazem o predicado formulado, diminuindo assim seu esforço na direção da interface.

Caso o sub-esquema seja maior que a área disponível para sua visualização, o recurso de rolagem será disponibilizado ao usuário por meio de barras horizontal e vertical permitindo assim sua visualização completa.

As ocorrências relativas às demais entidades que compõem o sub-esquema selecionado são mostradas, respeitando o predicado definido pelo usuário. Para a visualização os objetos complexos, utilizam-se as setas disponibilizadas nos *widgets* (*combo boxes*) que conterão tais atributos.

A janela do modo de visualização Gráfico é formada por um *container* que delimita a janela gráfica onde serão mostrados o sub-esquema. A janela do modo gráfico é apresentada na FIGURA 13.

Este modo de visualização é voltado para usuários novatos e intermitentes. Este modo requer que o usuário recorra poucas vezes às etapas anteriores, pois o modo de visualização mantém a consistência com a utilização dos elementos gráficos derivados das fases anteriores da interação facilitando também a visualização dos objetos complexos.

O modo gráfico aproveita a idéia da ferramenta SUPER [16], restringindo a navegação ao nível de ocorrências da resposta simplificando a visualização do conjunto de tais ocorrências para o usuário novato.

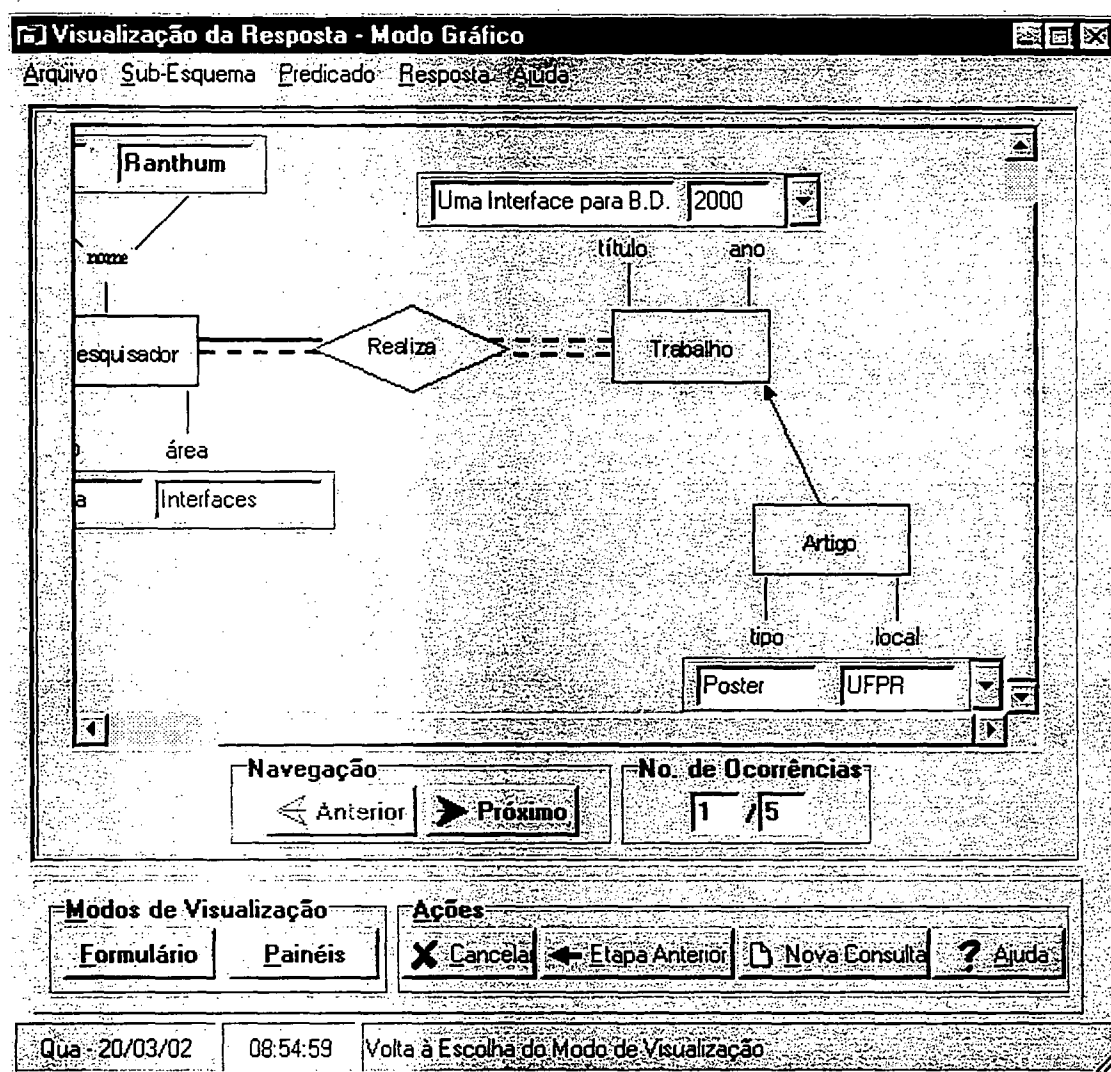


FIGURA 13 – JANELA DA RESPOSTA NO MODO GRÁFICO

Modo Painéis

Neste modo de visualização, o usuário terá as ocorrências exibidas na forma de painéis [1], dispostos de forma hierárquica, onde a hierarquia é dada pela inserção de painéis na forma horizontal.

Os painéis são dispostos de maneira que o usuário possa ver três ocorrências que satisfazem o predicado por vez. Para ter acesso às demais, este deverá usar os botões inseridos na caixa Navegação.

Os painéis exibem o resultado da consulta conforme o predicado formado pelo usuário. A entidade principal (raiz da consulta) destacada por meio do recurso de cor e de fonte diferenciadas. Nos painéis embutidos estarão as demais entidades

com suas ocorrências sempre respeitando o predicado formado pelo usuário. Para os objetos complexos, são disponibilizadas ao usuário barras de rolagem, que permitem a este verificar todas as ocorrências para aquela entidade.

A janela do modo de visualização Painéis é formada por um *container* que inclui os painéis que mostram as ocorrências. Para a navegação pelos atributos complexos, utilizam-se as barras de rolagem que serão apresentadas nos painéis que conterão tais atributos.

Os rótulos dos grupos de painéis são gerados a partir dos objetos escolhidos pelo usuário durante a seleção do sub-esquema. O objetivo é manter a consistência diminuindo assim o esforço cognitivo do usuário durante a interação, visto que, este modo de apresentação da resposta, não apresenta os elementos visuais que caracterizam as entidades no modelo ERC+ [37]. A janela do modo painéis está apresentada na FIGURA 14.

O modo painéis atende tanto ao grupo de usuários intermitentes como ao de usuários experientes. Este modo exige um maior poder de interpretação das respostas e de memorização dos elementos (entidades) selecionados durante as fases anteriores da interação, relativamente ao modo gráfico. Quanto à exploração das capacidades do modelo, este modo facilita a interpretação dos objetos complexos em relação ao modo formulário, embora exija um nível de abstração maior do que o modo gráfico.

Este modo de visualização é uma evolução da visualização da resposta no ambiente VIQUEN [19], no sentido de que aproveita a idéia da resposta estruturada fornecendo uma saída sem repetições de ocorrências.

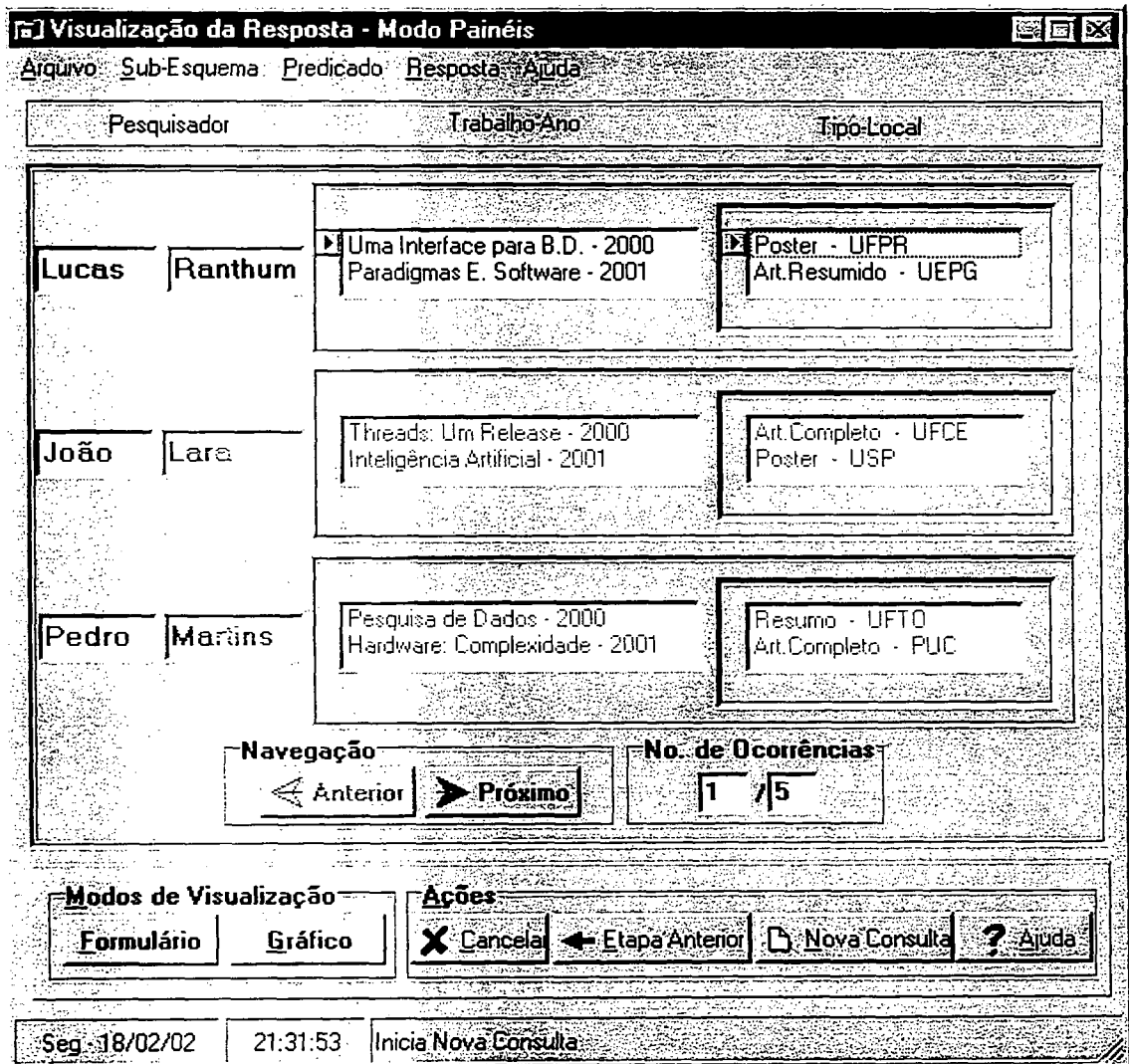


FIGURA 14 – JANELA DA RESPOSTA NO MODO PAINÉIS

5.7.3 Contribuições do Ambiente

O ambiente de interface proposto contribui de várias formas para a minimização do esforço do usuário na interpretação o resultado da consulta.

A barra de menus e os botões de pressão permitem ao usuário interagir com a aplicação e alcançar a meta desejada pro meio de qualquer um dos recursos. Caso a meta seja obter ajuda, o ambiente proporciona o alcance a partir de um item na barra de menus ou de um botão rotulado.

O usuário tem à sua disposição manuais *online* da aplicação que mostram como executar as tarefas disponíveis e também os *hints* que informam de maneira breve o significado/função de cada *widget* do ambiente.

A barra de título do sistema mostra ao usuário em qual modo da visualização da resposta o sistema se encontra; a barra de estado mostra mensagens de interação e de alerta.

As alterações são identificadas pelo usuário pela alteração da cor dos botões. Quando estão em cinza claro, dizem ao usuário que a função não está disponível no momento. Quando ativados ou disponíveis, sua cor torna-se cinza escuro. Estes recursos facilitam a percepção do estado do sistema pelo usuário.

Embora os erros sejam minimizados por meio de uma comunicação sistema-usuário projetada, a opção de desfazer está presente no ambiente em 2 níveis. Se o usuário ativar um modo de visualização que não deseje, pressionando o botão



, o ambiente retornará à situação de escolha do modo de visualização. Esta possibilidade se justifica pois remete à orientação no processo de seleção, que inclui a visualização de um exemplo associado a cada um dos modos disponíveis.

No nível da tarefa, o botão permite a volta à etapa de elaboração do predicado.

Finalmente os modos alternativos de visualização se complementam para atingir o objetivo de facilitar a percepção e a interpretação de objetos complexos, atendendo aos diversos perfis de usuários.

5.7.4 Quadro Comparativo

A seguir é apresentado um quadro que mostra uma análise comparativa das características das ferramentas de consultas (QBD*, SUPER e VIQUEN) analisadas neste trabalho no item referente à visualização da resposta. As características que diminuem o esforço cognitivo do usuário estão em destaque.

A coluna "INTERFACE PROPOSTA", se refere ao protótipo desenvolvido neste trabalho, o qual une as características favoráveis destas ferramentas agregando ainda características construídas a partir da revisão das bases teóricas consultadas no presente trabalho.

Ferramenta				
Característica	QBD*	SUPER	VIQUEN	INTERFACE PROPOSTA
Modos de Visualização	ÚNICO	FORMULÁRIO, GRÁFICO	ÚNICO	FORMULARIO, GRÁFICO, PAINÉIS.
Manutenção da Visualização Diagramática	NÃO	SIM (Modo Gráfico)	NÃO	SIM (Modo Gráfico)
Respostas Estruturadas	NÃO	SIM	SIM	SIM
FeedBack ao Usuário	NÃO	NÃO	NÃO	SIM
Auxílio ao Usuário	NÃO	NÃO	NÃO	SIM
Objeto “raiz” destacado	NÃO	NÃO	NÃO	SIM
Necessidade de Conhecimento do Esquema	SIM	NÃO	NÃO	NÃO
Exibição apenas no estilo “Tabela”	SIM	NÃO	NÃO	NÃO

QUADRO 3 – ANÁLISE COMPARATIVA DAS FERRAMENTAS DE CONSULTA

As características constantes das ferramentas foram analisadas da seguinte forma em relação à etapa da visualização da resposta:

Modos de Visualização: indica se a ferramenta apresenta mais de um modo visualização garantindo ao usuário maior flexibilidade.

Manutenção da Visualização Diagramática: indica se a ferramenta mostra as ocorrências da resposta juntamente com os elementos gráficos do sub-esquema mantendo a consistência com as fases anteriores à visualização da resposta (seleção do sub-esquema e formação do predicado).

Respostas Estruturadas: indica se a ferramenta apresenta a resposta de forma hierárquica, diminuindo o esforço cognitivo do usuário.

FeedBack ao Usuário: indica se a ferramenta oferece indicativos(retorno) ao usuário sobre o resultado de uma ação executada.

Auxílio ao Usuário: indica se a ferramenta disponibiliza ao usuário manuais e sistema de ajuda para solução de eventuais problemas de interação.

Objeto “raiz” em destaque: indica se a ferramenta mantém o usuário ciente do objeto escolhido como principal da consulta (raiz da consulta).

Necessidade de conhecimento do Esquema: indica se a ferramenta exige do usuário conhecimentos dos conceitos de banco de dados, o que dificultará a interação.

Exibição apenas no estilo “Tabela”: indica se a ferramenta exhibe as ocorrências na forma de tabelas, ou seja, uma seqüência de linhas e colunas.

6 CONCLUSÕES E TRABALHOS FUTUROS

6.1 CONCLUSÕES

Este trabalho apresentou um ambiente de interface para a ferramenta VIQUEN [18], construído sobre o modelo ERC+, capaz de diminuir o esforço do usuário na percepção e interpretação do resultado das consultas, realizadas a um banco de dados, preservando a forma de objetos complexos. Apesar de poder ser caracterizado como um ambiente de consulta visual, o VIQUEN não foi desenvolvido tendo-se a interface-usuário como foco principal.

Para avaliar a viabilidade da implementação do ambiente, foi desenvolvido e implementado um protótipo. O protótipo é um instrumento de avaliação formativa, isto é, avaliação feita anteriormente à implementação do sistema e que influencia o seu desenvolvimento [31]. Além disto, um protótipo permite a redução do tempo de desenvolvimento do sistema a ser implementado, possibilitando a correção de erros no início do processo de *design* como também que *designers* ofereçam um sistema de melhor qualidade para os usuários, reduzindo as chances destes não entenderem a mensagem a ser passada.

As abordagens cognitivas para o *design* de interface são de fundamental importância para a elaboração do modelo conceitual da aplicação buscando modelos do funcionamento da mente humana, porém deixam de considerar o amplo potencial comunicativo que as interfaces apresentam. Assim a construção deste protótipo baseou-se na Engenharia Semiótica [12] que tem como objetivo explorar esta característica de metacomunicação que os sistemas computacionais possuem e oferecer ao *designer* instrumentação que lhe permita *ensinar* quais soluções ele projetou para os problemas dos usuários.

Na Engenharia Semiótica o processo de *design* é visto como uma atividade semiótica, isto é, que inclui processos de produção de signos. Nestes processos existem as atividades de *formulação do conteúdo a ser comunicado* e a sua *representação em algum sistema expressivo*. Estas atividades devem ser guiadas por uma linguagem de especificação de maneira que se tenha uma ferramenta que auxilie o *designer* neste processo comunicativo.

A LEMD [21] foi de fundamental importância na construção deste protótipo pois permitiu estruturar os diversos tipos de mensagens que o *designer* necessitou enviar para comunicar ao 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 comando de função). Além disso, possibilitou especificar as diversas outras mensagens que auxiliam ao usuário no processo de utilização e aprendizado do sistema.

Utilizando a LEMD as mensagens foram especificadas de maneira abstrata independente de quais signos de interface seriam utilizados para concretiza-las. Regras de mapeamento semântico mostraram como a especificação abstrata da mensagem pode ser mapeada em *widgets* de ferramentas de interfaces gráficas. Este formalismo é complementar às linguagens de especificação tradicionais, podendo ser integrado às ferramentas de interfaces baseadas em *widgets*, normalmente utilizadas para interfaces gráficas.

6.2 TRABALHOS FUTUROS

A opção pela linguagem escolhida para implementação foi feita em virtude do conhecimento sobre a mesma. Entretanto ela não é a ideal visando a integração de todas as etapas de uma consulta a um banco de dados.

A continuação natural deste trabalho passa primeiramente pela integração dos subsistemas correspondentes às etapas envolvidas na consulta a um banco de dados (seleção do sub-esquema [6], elaboração do predicado [7] e apresentação do resultado). Esta integração será facilitada pela metodologia de desenvolvimento paralelo e conjunto dos três protótipos, prevendo-se apenas a eventual necessidade de pequenas alterações, para garantir a completa consistência da interface final.

Ainda como parte da integração esperada, o ambiente de interface deverá ser acoplado ao sistema de banco de dados de hipótese e ao ILOG (biblioteca de componentes visuais), utilizados na construção do ambiente VIQUEN [19].

Como uma possibilidade de pesquisa adicional, pode-se trabalhar ainda no estudo do armazenamento da resposta visando o seu reaproveitamento (reuso) em consultas subseqüentes.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ABITEBOUL, Serge, HULL, Richard, VIANU, Victor. **Foundations of databases** Addison-Wesley, 1995.
- [2] ANDERSEN, P. B. **A Theory of Computer Semiotics**, Cambridge University Press, Cambridge, 1990.
- [3] ANGELACCIO, Michele; CATARCI, Tiziana; SANTUCCI Giuseppe. **QBD*: A Fully Visual Query System. Journal of visual Languages and Computing**, Vol.1, No. 2, p.255-273, 1990. Disponível em:
<ftp://ftp.dis.uniroma1.it/pub/catarci/QBDJVLS.a.gz>
- [4] BARANAUSKAS, Maria Cecília; ROSSLER, Flávia; OLIVEIRA, Osvaldo Luiz. **Uma Abordagem Semiótica à Análise de Interface: Um Estudo de Caso.** Congresso de Interação Homem-Computador (1.: 1998 : Rio de Janeiro). Disponível em: www.inf.puc-rio.br/~ihc98/ataseletronicas .
- [5] BARBOSA, Simone D. J. SOUZA, Clarisse S de. **Making More Sence out of Users' Utterances**, IHC'99 – II Workshop sobre Fatores Humanos em Sistemas Computacionais, Campinas – SP, Outubro/99.
- [6] BASSI, Patrícia R. **Um Ambiente de Interface Visual para a Geração de Sub-Esquemas para uma Ferramenta de Consulta no Modelo ERC+**. Curitiba, 2002. Dissertação (Mestrado em Informática) – Setor de Ciências Exatas, Universidade Federal do Paraná.
- [7] BOSCARIOLI, Clodis. **Um Ambiente de Interface Visual para a Elaboração de Predicados para uma Ferramenta de Consulta sob o Modelo ERC+**. Curitiba, 2002. Dissertação (Mestrado em Informática) – Setor de Ciências Exatas, Universidade Federal do Paraná.

- [8] CATARCI, Tiziana. COSTABILE, Maria F. LEVIALDI, Stefano. BATINI, Carlo. **Visual Query Systems for Databases: a Survey**. Journal of Visual Language and Computing, volume 8, número 2, 1997, 215-260.
- [9] CATARCI, Tiziana; COSTABILE, Maria F.; CRUZ, Isabel F.; IOANNIDIS, Yannis and SHNEIDERMAN, B. **Data Models, Visual Representations, Methaphors: How to Solve the Puzzle?** IFIP WG2.6 3rd Working Conference on Visual Database Systems, p. 153-156, mar. 1995.
- [10] CATARCI, Tiziana; SANTUCCI Giuseppe. **Diagrmmatic Vs Textual Query Languages: A Comparative Experiment**. IFIP WG2.6 3rd Working Conference on Visual Database Systems, p. 57-73, mar. 1995.
- [11] DATE, C J. **Introdução a Sistemas de Banco de Dados**. Editora Campus. Rio de Janeiro- RJ. 1991.
- [12] DE SOUZA, Clarice S.. **The Semiotic Engineering of User Interface Languages**. International Journal of Man-Machine Studies 39, 1993, 753-773.
- [13] DE SOUZA, Clarisse S.; LEITE, Jair Cavalcanti; PRATES, Raquel Oliveira; BARBOSA, Simone D.J. **Projeto de Interfaces de Usuário: Perspectivas Cognitivas e Semióticas**. Disponível em: <http://peirce.inf.puc-rio.br/~clarisse/bookmarks.html>
- [14] DE SOUZA, Clarisse S.; PRATES, Raquel BARBOSA; Simone D. J. **A Method for Evaluating Software Communicability**. II Workshop sobre Fatores Humanos em Sistemas Computacionais, Campinas – SP, Outubro/99.

- [15] DE SOUZA, Clarisse S.; 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.
- [16] DENNEBOUY Yves; ANDERSSON Martin; AUDDINO Annamaria; YANN Dupont; FONTANA Edit; 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, p. 73-99, 1995.
- [17] DENNEBOUY, Yves. **Flexibility of visual languages for data manipulation**. IFIP WG2.6 3rd Working Conference on Visual Database Systems, p. 84-102, mar. 1995.
- [18] FAMILANT, M.E., DETWEILER, M.C. (1993). **Iconic reference: evolving perspectives and an organizing framework**. International Journal of Man-Machine Studies, V. 39, n. 4, p. 705-728.
- [19] GUEIBER, Ezequiel. **VIQUEN – Um Ambiente Interativo para Consulta Visual e Extração de Esquemas**. Curitiba, 2001. Dissertação (Mestrado em Informática) – Setor de Ciências Exatas, Universidade Federal do Paraná.
- [20] KUNTZ, Michel; MELCHERT, Rainer. **Pasta-3's graphical query language: Direct manipulation, cooperative Queries, full expressive power**. Proc. 15th International Conference on Very Large Data Bases, p. 97-105, 1989.
- [21] LEITE, J. C. **Modelos e Formalismos para a Engenharia Semiótica de Interface de Usuário**. Rio de Janeiro, 1998. Tese (Doutorado em Informática). Pontifícia Universidade Católica do Rio de Janeiro.

- [22] LEITE, J. C.; DE SOUZA, C. S. **Uma Linguagem de Especificação para a Engenharia Semiótica de Interfaces de Usuário**, II Workshop sobre Fatores Humanos em Sistemas Computacionais, Campinas – SP, Outubro/99.
- [23] MARTINS, Isa Haro. **Um instrumento de Análise Semiótica para Linguagens Visuais de Interfaces**. . Rio de Janeiro, 1998. Tese (Doutorado em Informática). Pontifícia Universidade Católica do Rio de Janeiro.
- [24] MARTINS, Isa Haro; DE SOUZA, Clarisse Sieckenius. **Uma Abordagem Semiótica na Utilização dos Recursos Visuais em Linguagens de Interface**. Congresso de Interação Homem-Computador (1.: 1998 : Rio de Janeiro). Disponível em: www.inf.puc-rio.br/~ihc98/ataseletronicas .
- [25] MURRAY, Norman; NORMAN Paton; GOBLE Carole. **A Framework for Describing Visual Interfaces to Databases**. Manchester, 1998. Disponível em: <http://www.cs.man.ac.uk/~murrayn>
- [26] NAVATHE, Shamkant B., ELMARSI, Ramez A. **Fundamentals of Database Systems**. Addison-Wesley, 1999.
- [27] NIELSEN, J. **Noncommand User Interfaces**, Communications of the ACM, vol. 36, abril de 1993.
- [28] NORMAN, D.A. **Cognitive Engineering**. In: D.A. Norman & S.W. Draper, eds. User Centered System Design. Hillsdale,N.J, 1986. Lawrence Erlbaum.
- [29] OLIVEIRA, Osvaldo Luiz de. BARANAUSKAS, M. Cecília Calani. **A Semiótica e o Design de Software**, Relatório Técnico IC-98-09, Abril/1998.

- [30] OLIVEIRA, Osvaldo Luiz de. BARANAUSKAS, M. Cecília Calani. **Interface Entendida Como um Espaço de Comunicação**. Atas do II Workshop sobre Fatores Humanos em Sistemas Computacionais. Campinas/SP. 1999.
- [31] PREECE, J.; Rogers, Y.; Sharp, E.; Benyon, D.; Holland, S.; Carey, T. **Human-Computer Interaction**. Addison-Wesley, 1994
- [32] PRADO, A. B.; BARANAUSKAS, M. C. C. **Avaliando a Meta-comunicação Designer-Usuário de Interface**. Atas do III Workshop sobre Fatores Humanos em Sistemas Computacionais. IHC'2000, Unisinos/RS, 2000.
- [33] 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.
- [34] PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995.
- [35] RODACKI, Andréa. **Aplicação de Estratégias de Integração de Banco de Dados: um Estudo de Caso**. Curitiba 2000. Dissertação (Mestrado em Informática) – Setor de Ciências Exatas, Universidade Federal do Paraná.
- [36] SHNEIDERMAN, B. **Designing the User Interface**, 2nd.edition: AD.Wesley, 1993.
- [37] SPACCAPIETRA, Stefano; PARENT, Christine; SUNYE, Marcos; YETONGNON, Kokou; LEVA, Antonio Di. **ERC + : An Object + Relationship Paradigm for Database Applications**. Readings in Object-Oriented Systems, D. Rine (Ed.), IEEE Press, 1995

APÊNDICE 1 – AUXÍLIO AO USUÁRIO

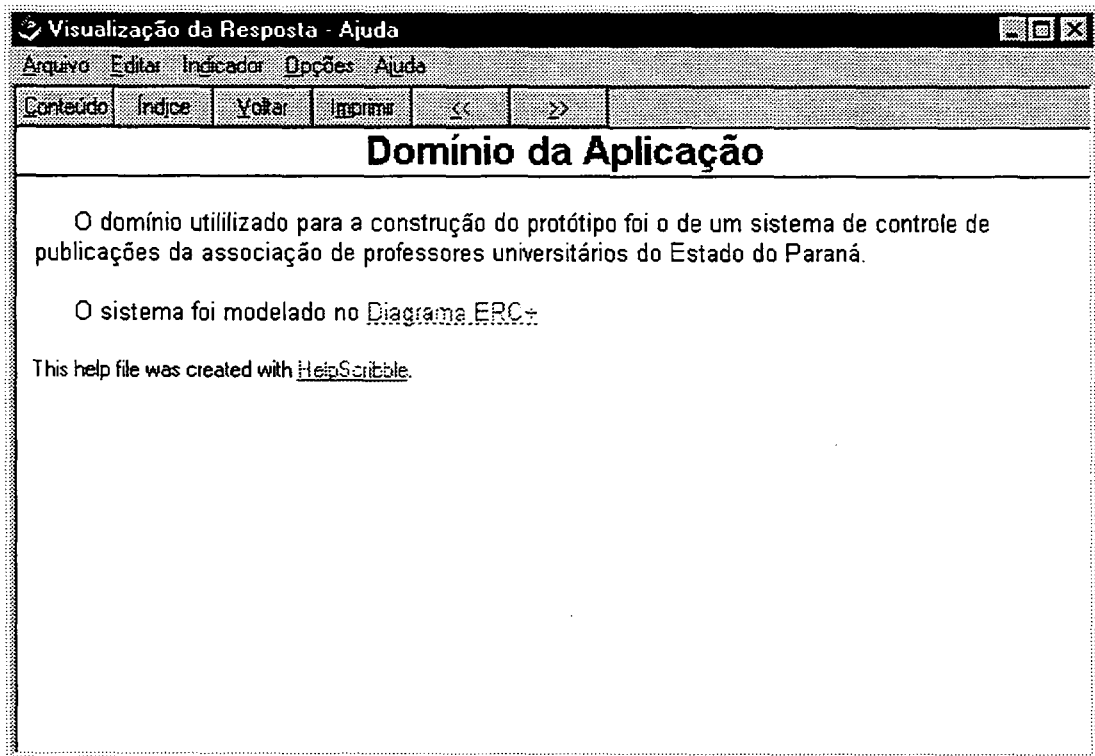
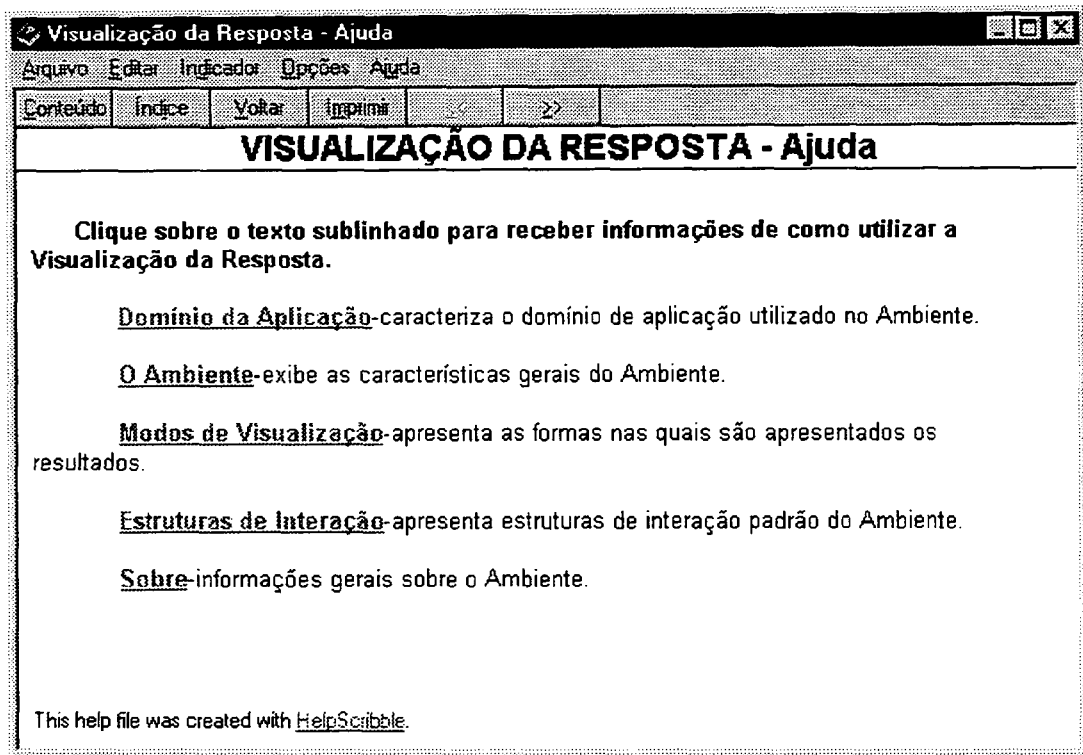
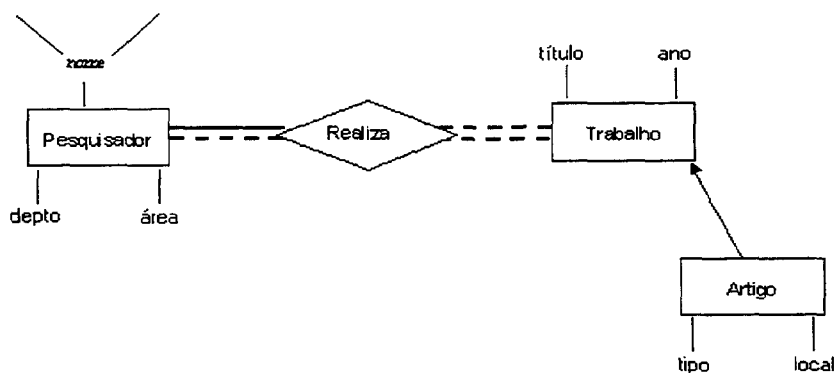


Diagrama ERC+

O modelo Entidade Relacionamento Complexo (ERC+), é um modelo que adiciona ao paradigma Entidade-Relacionamento conceitos do modelo comportamental da orientação a objetos.

O modelo ERC+ é chamado ERC+ devido ao seu significado: ER para Objetos Complexos (o + denota o enriquecimento do modelo ERC básico).



This help file was created with [HelpScribble](#).

Visualização da Resposta - Ajuda [Ícones de janela]

Arquivo Editar Inicializador Opções Ajuda

Conteúdo Índice Voltar Imprimir << >>

O Ambiente

O ambiente disponibiliza três formas para visualização do resultado da consulta proporcionando maior flexibilidade ao usuário.

O ambiente foi desenvolvido em uma ferramenta padrão Windows, construído em uma interface gráfica: *GUI (Graphical User Interface)* no estilo *WIMP (Windows, Icons, Menus e Point Devices)*.

This help file was created with [HelpScribble](#).

Visualização da Resposta - Ajuda

Arquivo Editar Indicador Opções Ajuda

Conteúdo Índice Voltar Imprimir << >>

Modos de Visualização

O usuário poderá optar pela forma com que ele deseja que sejam apresentados os resultados. Estas formas foram denominadas: **Modo Formulário**, **Modo Gráfico** e **Modo Painéis**.

Uma vez feita a opção, um botão com o nome do modo escolhido será ativado. Além disso, o usuário terá uma visão prévia de como serão apresentados os dados naquele modo.

Visualizar Resposta

Visualizar Resposta

Visualizar Resposta

This help file was created with [HelpScribble](#).

Modo Formulário

O modo de visualização Formulário, apresentará ao usuário as ocorrências através de grades (*grids*). A entidade principal escolhida na etapa da seleção do sub-esquema (raiz da consulta) será a primeira a aparecer no formulário e estará destaca com cor de fonte azul.

This help file was created with [HelpScribble](#).

Modo Gráfico

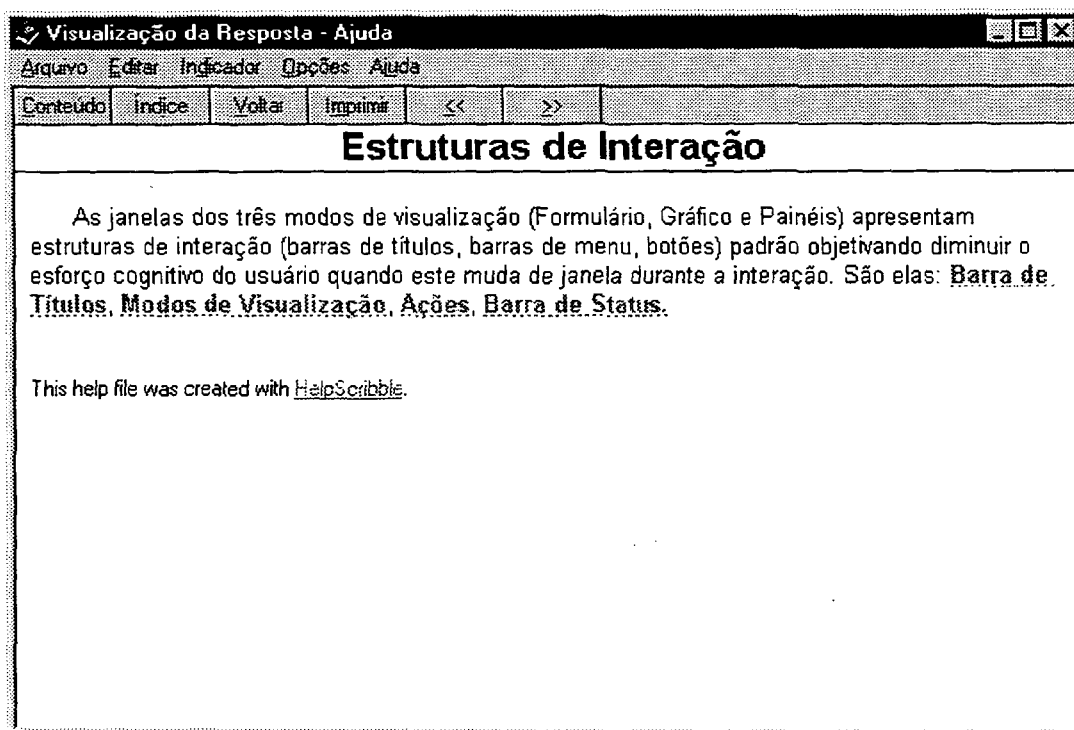
O modo de visualização Gráfico permite ao usuário ter as ocorrências ligadas ao sub-esquema selecionado, isto é, os símbolos gráficos (notação do modelo) serão derivados da construção diagramática do modelo ERC+.

This help file was created with [HelpScribble](#).

Modo Painéis

O modo de visualização Gráfico permite ao usuário ter as ocorrências dispostas na forma de painéis. Os painéis foram dispostos de maneira que o usuário possa ver três ocorrências que por vez. A entidade principal (raiz da consulta) será destacada com cor de fonte azul.

This help file was created with [HelpScribble](#).



Barra de Títulos e Barra de Menus

Na barra de títulos é indicado o modo de Visualização da Resposta em execução.

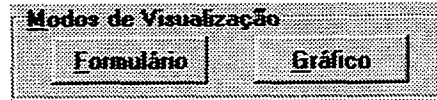
Na barra de menus, o usuário encontra as opções disponíveis no ambiente.



This help file was created with [HelpScribble](#).

Modos de Visualização

Estes botões permitem ao usuário alternar entre os modos de visualização da resposta.



This help file was created with [HelpScribble](#).

Ações

Disponibiliza ao usuário as ações que este poderá executar durante a interação.



This help file was created with [HelpScribble](#).

Cancelar

Permite ao usuário retornar à janela de seleção do modo de exibição.

This help file was created with [HelpScribble](#).

Etapa Anterior

Permite ao usuário cancelar a etapa corrente. Este cancelamento faz com que o usuário volte para a etapa de elaboração do predicado para realizar nova consulta.

This help file was created with [HelpScribble](#).

Nova Consulta

Permite ao usuário iniciar nova Consulta. Em qualquer um dos modos de visualização que o usuário esteja ele poderá reiniciar todas as fases da consulta.

This help file was created with [HelpScribble](#).

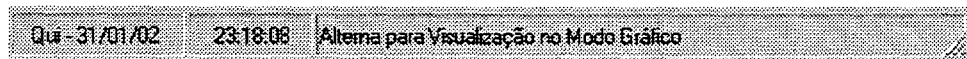
Ajuda

Permite ao usuário obter ajuda sobre a etapa corrente. Caso esteja na janela inicial, a ajuda se dará sobre a aplicação como um todo. Caso esteja em uma das janelas do modo de visualização, a ajuda será sobre àquele modo.

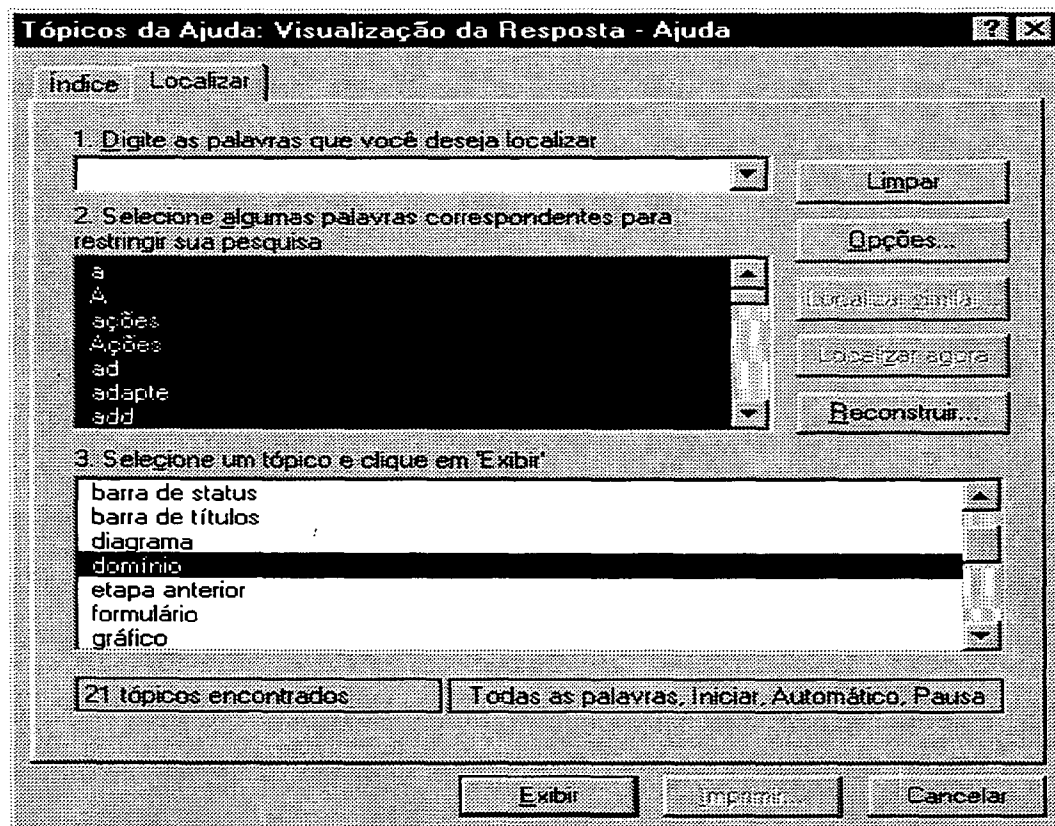
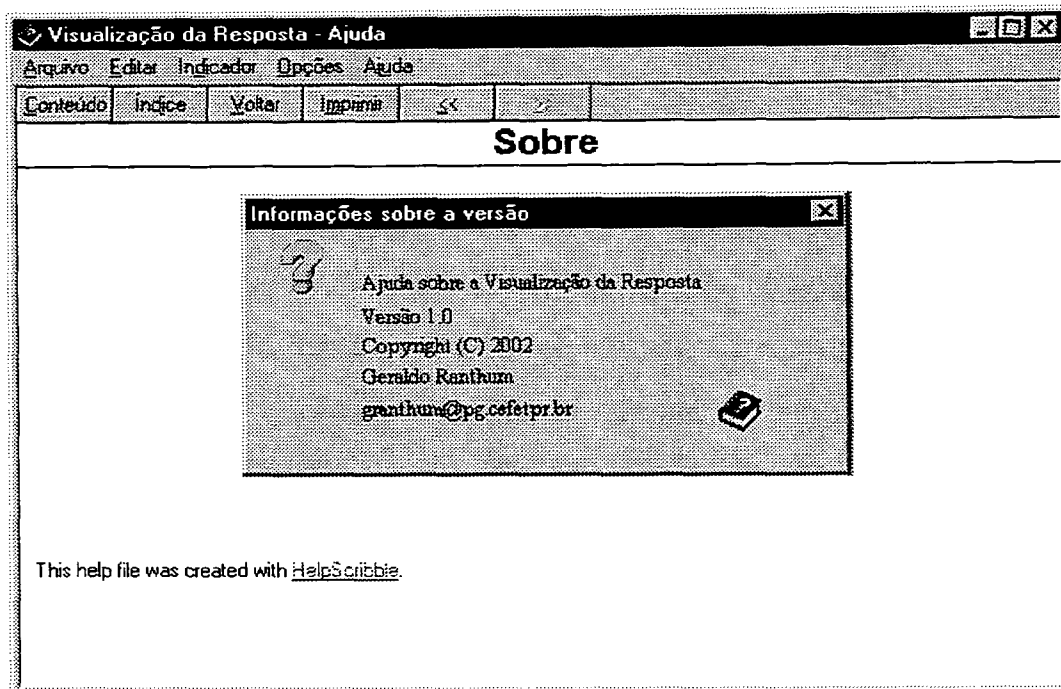
This help file was created with [HelpScribble](#).

Barra de *Status*

A barra de estado do sistema (*status bar*) mostra, além da data e hora corrente, descrições sobre a função de cada componente de interação presentes naquela janela.



This help file was created with [HelpScribble](#).



APÊNDICE 2 – LEMD DA INTERFACE PROPOSTA

II Início Visualização Resposta

Command-Message Visualização da Resposta for Application-function

Resposta

Join {

Join {

Sequence {

Combine {

Text "Modos Disponíveis"

Text "Exemplo"

Select Set-of *Formulário, Gráfico, Painéis*

View Information-of *Exemplo*

Activate Start for Application-Function *Visualizar Resposta*

}

}

Join {

Select {

View "Ações"

Activate Start for Application-Function *Etapa Anterior*

Activate Start for Application-Function *Nova Consulta*

Activate Start for Application-Function *Ajuda*

}

}

Join {

Combine {

View Information-of "Data"

View Information-of "Hora"

View Information-of "Mensagens"

}

}

}

Command-Message Visualização da Resposta for Application-function

Formulário

Join {

View "Visualização da Resposta - Modo Formulário"

Sequence {

Combine {

View Information-of "Pesquisador"

View Information-of "Trabalho"

View Information-of "Artigo"

}

```

Combine {
    View "Navegação"
    Activate Start for Application-Function Anterior
    Activate Start for Application-Function Próximo
    View Information-of No. de Ocorrências
}
}

Join {
Select {
    View "Modos de Visualização"
    Activate Start for Application-Function Gráfico
    Activate Start for Application-Function Painéis
}
Select {
    View "Ações"
    Activate Start for Application-Function Cancelar
    Activate Start for Application-Function Etapa Anterior
    Activate Start for Application-Function Nova consulta
    Activate Start for Application-Function Ajuda
}
}
Join {
    Combine{
        View Information-of "Data"
        View Information-of "Hora"
        View Information-of "Mensagens"
    }
}
}

Command-Message Visualização da Resposta for Application-function
Gráfico
Join {
    View "Visualização da Resposta - Modo Gráfico"
    Sequence {
        Combine {
            View Information-of "Sub-Esquema"
        }
    }
    Combine {
        View "Navegação"
        Activate Start for Application-Function Anterior
        Activate Start for Application-Function Próximo
        View Information-of No. de Ocorrências
    }
}
Join {

```

```

Select {
  View "Modos de Visualização"
  Activate Start for Application-Function Formulário
  Activate Start for Application-Function Painéis
}
Select {
  View "Ações"
  Activate Start for Application-Function Cancelar
  Activate Start for Application-Function Etapa Anterior
  Activate Start for Application-Function Nova consulta
  Activate Start for Application-Function Ajuda
}
}
Join {
  Combine{
    View Information-of "Data"
    View Information-of "Hora"
    View Information-of "Mensagens"
  }
}
}

```

Command-Message *Visualização da Resposta for Application-function*
Painéis

```

Join {
  View "Modo Painéis"
  Sequence {
    Combine {
      View Information-of "Pesquisador"
      View Information-of "Trabalho-Ano"
      View Information-of "Tipo-Local"
    }
  }
  Combine {
    View "Navegação"
    Activate Start for Application-Function Anterior
    Activate Start for Application-Function Próximo
    View Information-of No. de Ocorrências
  }
}
Join {
  Select {
    View "Modos de Visualização"
    Activate Start for Application-Function Formulário
    Activate Start for Application-Function Gráfico
  }
}

```

```
Select {  
  View "Ações"  
    Activate Start for Application-Function Cancelar  
    Activate Start for Application-Function Etapa Anterior  
    Activate Start for Application-Function Nova consulta  
    Activate Start for Application-Function Ajuda  
  }  
}  
Join {  
  Combine{  
    View Information-of "Data"  
    View Information-of "Hora"  
    View Information-of "Mensagens"  
  }  
}  
}
```

II Término Visualização Resposta

APÊNDICE 3 – STAG DA INTERFACE PROPOSTA

Declaração de Tipos e Instâncias

1. Tipos de Dados do Sistema

primitiva
 primitiva-interação
 primitiva-composta
 plano-interação

1. Instâncias de Dados do Sistema

primitiva: entidade, relacionamento, cardinalidade (opcional monovalorado 0:1), cardinalidade (monovalorado e mandatório 1:1), cardinalidade (opcional e multivalorado 0:N), cardinalidade (mandatória e multivalorado 1:N), atributo, atributo_complexo
 primitiva-interação: botão, barra de rolagem, caixa de listagem
 primitiva-composta: sub-esquema
 plano-interação: janela_inicial, formulário, gráfico, painéis

1. Tipos de Operações

operações-edição

1. Instâncias de Operações

operações-edição: selecionar, alternar.

1. Tipos de Variáveis de Estados de Execução do Sistema

estado-seleção

1. Instâncias de Variáveis de Estados de Execução do Sistema

estado-seleção: ativo, inativo

1. Relações de Sobreposição

primitiva <-- estado-seleção

1. Tipos de Expressão do Sistema

forma
 forma-composta
 diagrama
 cor
 texto

1. Instâncias de Expressão do Sistema

forma: retângulo, losango, linhas cheias e/ou tracejadas, palavra
 forma-composta: botão, botão_sombreado, janela_rolagem, janela
 diagrama: retângulo, losango, linhas cheias e/ou tracejadas, palavra
 cor: cinza-escuro, cinza-claro, preto, amarelo, azul
 texto: menu-hierárquico, rótulo

1. Tipos do Conteúdo do Usuário

intenção

1. Instâncias do Conteúdo do Usuário

intenção: indicar, movimentar, ativar_janela, posicionar, retornar

1. Tipos de Expressão do Usuário

sinal

1. Instâncias de Expressão Manual

sinal: clique-botão-esquerdo, pressão-e-arrasto, parar-com-mouse.

Mapeamento

1. Mapeamento de Tipos do Sistema

primitiva <=> forma

primitiva-interação <=> cor

estado-seleção <=> cor <> forma-composta

plano-interação <> cor

1. Mapeamento de Instâncias do Sistema

primitiva (entidade) <=> forma (retângulo)

primitiva (relacionamento) <=> forma (losango)

primitiva (cardinalidade (monovalorado e mandatório 1:1)) <=> forma (linha simples contínua)

primitiva (cardinalidade (opcional monovalorado 0:1)) <=> forma (linha simples tracejada)

primitiva (cardinalidade (opcional e multivalorado 0:N)) <=> forma (linha dupla tracejada)

primitiva (cardinalidade (mandatória e multivalorado 1:N)) <=> forma (linha tracejada e linha contínua)

primitiva (atributo) <=> forma (palavra)

primitiva (atributo_complexo) <=> forma (palavra)

diagrama <=> forma (retângulo, losango, linhas cheias e/ou pontilhadas, palavra)

primitiva-interação (raiz) <=> cor (amarelo)

primitiva-interação (diagrama_subesquema) <=> cor (preto)

estado-seleção (ativo) <=> cor (preto) <> forma-composta (botão)

estado-seleção (ativo) <=> cor (cinza-escuro) <> forma-composta (botão)

estado-seleção (ativo) <=> cor (azul) <> forma-composta (caixa)

estado-seleção (inativo) <=> cor (cinza-claro) <> forma-composta (botão)

estado-seleção (inativo) <=> nulo

plano-interação (janela_inicial) <=> cor (cinza)

plano-interação (formulário) <=> cor (cinza)

plano-interação (gráfico) <=> cor (cinza)

plano-interação (painéis) <=> cor (cinza)

1. Mapeamento de Tipos do Usuário

intenção <=> sinal

1. Mapeamento de Instâncias do Usuário

intenção (indicar) <=> sinal (clique-botão-esquerdo)

intenção (movimentar) <=> sinal (pressão-e-arrasto)

intenção (ativar_janela) <=> sinal (clique-botão-esquerdo)

intenção (posicionar) <=> sinal (parar-com-mouse)

intenção (retornar) <=> sinal (clique-botão-esquerdo)

Dicionário

1. Dicionário de Tarefas Básicas

Selecionar_layout_visualização [operação = indicar, objeto=botão, critério único]

Navegar [operação = movimentar, objeto=barra de rolagem, critério único]

Selecionar_dado [operação = selecionar, objeto=caixa de listagem, critério único]

Alternar [operação = indicar, objeto=botão, critério único]

Retornar [operação = retornar, objeto=botão, critério único]

Gramática

1. Regras Gramaticais

R1. Selecionar_layout_visualização [Critério = único] →

Plano-Interação (janela_inicial)+

Intenção (indicar)+

Estado-seleção (ativo)

R2. Navegar [Critério = único] →

Intenção (navegar)+

Estado-seleção (ativo)

R3. Alternar [Critério = único] →

Plano-Interação(plano)+

Intenção (alternar)+

Estado-seleção (ativo)

R4. Retornar [Critério = único] →

Primitiva-Composta (sub-esquema)+

Intenção (retornar)+

Estado-seleção (ativo)
