CLODIS BOSCARIOLI

UM AMBIENTE DE INTERFACE VISUAL PARA A ELABORAÇÃO DE PREDICADOS PARA UMA FERRAMENTA DE CONSULTAS A BANCO DE DADOS SOB O MODELO ERC+

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

Orientadora: 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 *Clodis Boscarioli*, avaliamos o trabalho intitulado, "Um Ambiente de Interface Visualpara a Elaboração de Predicados para uma Ferramenta de Consultas a Banco de Dados sob o Modelo ERC+", cuja defesa foi realizada no dia 07 de maio 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 do candidato.

Curitiba, 07 de maio de 2002.

Profª. Dra. Laura Sanchez Garcia
PGINF/UFPR - Orientadora

Profª. Dra. Heloisa Vieira da Rocha
IC/UNICAMP

Prof. Dr. Marcos Sfair Sunye
DINF/UFPR

Esta dissertação é dedicada aos meus pais Laércio e Clélia que me deram toda a base educacional para que eu pudesse aqui chegar. Em especial a minha noiva e futura esposa Lucia Specia pela infindável compreensão, paciência e apoio.

AGRADECIMENTOS

A Deus por permitir-me a vida e ser sempre meu refúgio e proteção.

A amada minha família (Clélia, Laércio, Cássia, Cleiton, Antonio, Camila e Flávio) pelo amor, pelos bons momentos e pela crença na minha capacidade.

A minha noiva Lucia pelas sugestões no trabalho.

Aos meus amigos Benefrancis, Fabíola, Lucio, Maria do Socorro e Laudemir pelas palavras de estímulo nos momentos de cansaço.

Aos meus colegas de trabalho da UNIOESTE, em especial a minha amiga Sarajane.

A "turma do carro": Ezequiel, Mônica, Geraldo e Simone, pela companhia durante o período das disciplinas e pelas risadas das viagens.

Aos amigos que ganhei pelo Mestrado: Roberta, David e Patrícia, pela presença e carinho.

A coordenação do Curso de Tecnologia em Informática do CEFET-PG por disponibilizar, sempre que necessário, a sua infra-estrutura para esta pesquisa.

A Laura, que além de orientadora tornou-se uma amiga, pelo crédito e atenção despendida.

Não poderia deixar de agradecer também a todos os alunos que conviveram comigo neste período, compreendendo minhas ocasionais ausências.

A todos que fazem parte da minha história e que, de uma forma ou de outra, contribuíram para mais este objetivo fosse atingido, o meu sincero "MUITO OBRIGADO".

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ANSI – *American National Standards Institute*
BD – Banco de Dados
E-R – Entidade-Relacionamento
ERC+ – Entidade Relacionamento Complexo Estendido
GUI – *Graphical Visual Interface*
IHC – Interface Humano-Computador
LEMD – Linguagem de Especificação da Mensagem do *Designer*
LV – Linguagem Visual
MUA – Modelo de Usabilidade da Aplicação
OLTP – *On Line Transaction Processing*
QBD* – *Query by Diagram*
QBE – *Query by Example*
SGBD – Sistema Gerenciador de Banco de Dados
SQL – *Structured Query Language*
STAG – *Semiotic Task-Action Grammar*
UCSD – *User Centered System Design*
VIQUEN – *Visual Query Environment*
VQS – *Visual Query System*
WIMP – *Windows, Icons, Menus e Pointing Devices*

RESUMO

Uma dificuldade clássica no acesso a bancos de dados tem sido causada pela necessidade de utilização, pelo usuário final, de linguagens de consulta textuais. Com o crescimento e a diversificação do conjunto de usuários finais de sistemas de banco de dados, surge a necessidade de interfaces que facilitem o processo de interação entre o homem e o computador na formulação de consultas. Este trabalho aborda sistemas visuais de consulta a banco de dados, mais especificamente, a etapa de elaboração de predicados, à luz da Engenharia Semiótica. Faz uma análise sobre ferramentas visuais de consulta e apresenta a utilização de dois formalismos de orientação para a construção de interfaces, a LEMD (Linguagem de Especificação da Mensagem do *Designer*), que auxilia na definição da mensagem sobre o modelo de usabilidade e a STAG (*Semiotic Task-Action Grammar*), que objetiva identificar e representar tarefas do usuário de forma categorizada. Apresenta um protótipo de interface desenvolvido para uma ferramenta de consultas a banco de dados baseada no modelo ERC+, e, por fim, discute as contribuições do trabalho e as metas futuras de pesquisa.

Palavras-Chave: Engenharia Semiótica, Interação Humano-Computador, Sistemas Visuais de Consulta.

ABSTRACT

A classical difficulty in the database access has been caused by the necessity of utilization by the end-user of textual consult languages. The growth and the diversification of database systems end-user groups arises the need of interfaces that make easer the interaction process between the user and the computer during the creation of queries. This work deals visual query systems for database, more specifically, the stage of predicate elaboration, under the Semiotic Engineering view. It makes an analysis about visual query tools and presents the use of two formalisms of orientation to the interface construction: DMSL (Designer's Message Specification Language), that helps the message definition about the usability model; and STAG (Semiotic Task-Action Grammar), that aims to identify and to represent user tasks in a categorized way. It presents, as well, a prototype of interface developed for a database query tool based on the ERC+ model. Finally, it discusses the work contributions and the goals for future research.

Keywords: Semiotic Engineering, Human-Computer Interaction, Visual Query Systems

1 INTRODUÇÃO

1.1 HISTÓRICO E PROBLEMA

Os bancos de dados têm, ao longo dos anos, alcançado um espaço cada vez maior nas organizações. Por outro lado, tem aumentado também, a preocupação com a manipulação eficiente das informações contidas nestas bases de dados, evidenciando então, a necessidade de linguagens de consulta voltadas ao usuário final e não somente ao desenvolvedor de aplicações. Com o crescimento e a diversificação dos usuários dos bancos de dados, tanto novatos quanto experientes, tem se acentuado o desenvolvimento de interfaces baseadas em diferentes princípios, cujo objetivo principal é o de facilitar o processo de interação entre o homem e o computador (CATARCI *et al.*, 1996a). Tradicionalmente, uma forma de fazer com que os usuários obtivessem conhecimento necessário sobre um sistema e sua utilização, era por meio de materiais escritos (manuais de usuários) e *help* de *software* (nem sempre disponíveis e eficazes), sem utilizar-se da interface para transmissão deste conhecimento.

Uma dificuldade clássica no acesso a bancos de dados tem sido causada pela necessidade de utilização, pelo usuário final, de linguagens de consulta textuais (tais como SQL - *Structured Query Language*). Uma linguagem de consulta representa, basicamente, um conjunto de operadores que foram formalmente definidos e adequadamente compostos. A maioria das linguagens de consulta desenvolvidas apresentam uma sintaxe que se assemelha a uma linguagem de programação, onde a consulta é expressa por uma *string* linear, na qual ambos operadores e operandos são denotados por termos. Além disso, a sintaxe dos operadores é geralmente complexa para aprendizado e a interação entre o usuário e o sistema é feita por meio de um teclado que permite o envio de texto (BATINI *et al.*, 1992). Enquanto a alternativa das interfaces em linguagem natural traz consigo a limitação da disparidade entre o escopo lingüístico do sistema e do usuário, transposta pela solução de interfaces por menus, esforços têm sido feitos no sentido de proporcionar outras alternativas de linguagens de interface, icônicas e de

manipulação direta, estas últimas identificadas (por sua co-ocorrência freqüente) com as interfaces gráficas. No âmago da exploração das interfaces gráficas e de manipulação direta para consultas a banco de dados, surgem as linguagens de consulta visuais (*Visual Query Languages*), consideradas uma evolução em relação às linguagens de consulta textuais existentes pois tentam suprir as dificuldades de usabilidade que os usuários encontram nos ambientes de consultas textuais.

Por sua vez, estas tentativas não têm alcançado os resultados esperados, na medida em que consultas complexas são difíceis de compor por falta de percepção, pelo usuário, das regras construtivas da linguagem visual.

No acesso a banco de dados, estes problemas adquirem importância especial. Embora consultas simples, como *"Quero saber o nome dos empregados que trabalham no projeto X"* sejam fáceis de elaborar por meio de linguagens visuais, ao se aumentar a complexidade e tentar formular consultas como *"Quero saber quais professores que lecionam a disciplina X, não lecionam a disciplina Y e estão envolvidos no projeto Z"*, o usuário pode perder-se no processo, por não perceber as regras de articulação associadas à linguagem, ou ainda, por não estar familiarizado às etapas para elaboração de uma consulta.

Apesar das facilidades associadas às interfaces de manipulação direta, elas não são, em si mesmas, infalíveis (SHNEIDERMAN, 1993). Assim como qualquer outro estilo de interação, elas necessitam cumprir critérios gerais, como a consistência, sem os quais elas não obedecem aos requisitos mínimos de uma boa interface. Por outro lado, estas têm aspectos negativos que lhes são inerentes, como a dificuldade em expressar nuances de significado, processos dinâmicos e conceitos abstratos, associados, por definição, a uma linguagem de comandos.

Apesar dos diversos esforços realizados no sentido de minimizar a dificuldade no uso de linguagens de consulta a bancos de dados, estas características das interfaces gráficas e de manipulação direta são responsáveis por não existirem ainda, resultados completamente satisfatórios.

Nesta direção, busca-se aumentar o poder semântico dos banco de dados, propondo-se extensões ao modelo Entidade-Relacionamento (E-R) amplamente difundido, como o modelo ERC+ (*SPACCAPIETRA et al.*, 1995a), (SPACCAPIETRA

et al., 1995b), de forma a aproximar ainda mais os modelos do mundo real com o percebido pelos usuários. Segundo (SILBERSCHATZ *et al.*, 1999), uma interface simples e fácil de usar exige uma correspondência um-para-um entre a noção de um objeto para o usuário e a noção de um item de dados para o banco de dados.

Este trabalho discute alguns ambientes que suportam tal paradigma, em particular, o protótipo VIQUEN (GUEIBER, 2001), que se constitui no ponto de partida da pesquisa aqui relatada. No desenvolvimento desse protótipo, não houve preocupação especial com a interface, sendo esta, portanto, objeto principal do presente trabalho.

1.2 OBJETIVOS DA DISSERTAÇÃO

Esta dissertação tem por objetivo geral desenvolver um ambiente de interface-usuário para elaboração de consultas na ferramenta VIQUEN.

Entre os objetivos específicos, têm-se:

- Analise de ferramentas para consulta visuais;
- Avaliação do ambiente VIQUEN baseada na Engenharia Semiótica;
- Proposta de um ambiente semiótico de interface-usuário para o processo de elaboração de consultas sob o modelo ERC+, à luz da Engenharia Semiótica.

1.3 ESCOPO E LIMITAÇÕES

O principal foco deste trabalho é o *design* de interface para banco de dados, visando o aumento da usabilidade e comunicabilidade de ambientes de consultas centrados no usuário.

Leva-se em consideração interfaces de usuário gráficas *GUI* (*Graphical User Interface*) no estilo *WIMP* (*Windows, Ícons, Menus e Pointing Devices*), bem como usuários familiarizados a este estilo.

Embora o processo de especificação de uma consulta visual passe por etapas relacionadas (seleção do sub-esquema, elaboração do predicado e apresentação do

resultado da consulta) esta pesquisa aborda a fase de elaboração do predicado, ou seja, a maneira como o usuário interage com o ambiente para escolher os atributos que comporão a resposta, bem como para o estabelecimento de critérios a serem respeitados na execução da consulta.

1.4 ESTRUTURA DA DISSERTAÇÃO

Este trabalho é estruturado da seguinte forma:

O capítulo um apresentou os principais obstáculos enfrentados pelo usuário na interação de banco de dados e, na seqüência, trouxe os objetivos, escopo e limitações e a estrutura deste trabalho.

No capítulo dois é apresentado um embasamento conceitual, descrevendo Engenharia Cognitiva e a Engenharia Semiótica e dois métodos semióticos para a análise de interfaces: a LEMD (Linguagem de Especificação da Mensagem do *Designer* (LEITE, 1998) e a STAG (*Semiotic Task-Action Grammar*) (MARTINS, 1998) na área de Interação Humano-Computador (IHC); e é discutido o conceito de Modelo de Dados, descrevendo o modelo E-R e sua extensão, o modelo ERC+. A compreensão destes conceitos faz-se necessária para o acompanhamento do restante do trabalho.

Questões sobre a interface-usuário para banco de dados, suas características e suas necessidades são tratadas no capítulo três.

O capítulo quatro descreve linguagens de consulta (textuais e visuais), evidenciando sua evolução junto à crescente demanda dos bancos de dados e, ainda, analisa algumas ferramentas destinadas ao usuário final, disponíveis na academia e no mercado.

No capítulo cinco, uma interface embasada na Engenharia Semiótica para a elaboração de consulta no protótipo VIQUEN é apresentada, juntamente com exemplos de utilização.

Por fim, o capítulo seis relata as conclusões sobre o trabalho desenvolvido e apresenta propostas para trabalhos futuros, dando continuidade à pesquisa realizada.

Como Apêndices, seguem a LEMD – Apêndice A, a STAG – Apêndice B e o auxílio completo, como Apêndice C. A Álgebra ERC+ é apresentada como Anexo A.

2 ABORDAGENS TEÓRICAS DA IHC

2.1 A IHC E AS ABORDAGENS TEÓRICAS QUE A SUSTENTAM

A interação Humano-Computador é uma área de estudos relativamente recente, mas com desenvolvimento surpreendente, abrangendo os mais diferentes tipos de sistemas computacionais.

O objetivo principal da IHC é a satisfação do usuário no processo de interação com sistemas computacionais. Com este fim, é feita a análise dos fatores envolvidos no desenvolvimento de interfaces, bem como do impacto destas no meio onde atuam. O usuário é o foco principal, para o qual direcionam-se todos os estudos e desenvolvimentos.

Para (SOUZA *et al.*, 1999), a interação é um processo que engloba as ações do usuário sobre a interface de um sistema, e suas interpretações sobre as respostas reveladas por esta interface. Esta visão é mostrada na Figura 1.

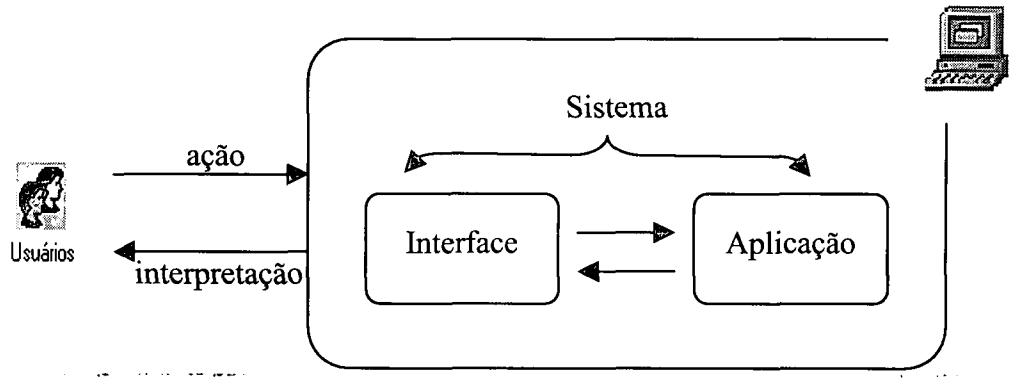


FIGURA 1 - PROCESSO DE INTERAÇÃO HUMANO-COMPUTADOR
FONTE - (SOUZA *et al.*, 1999)

A partir dessas definições podemos dizer que a interface é um sistema de comunicação, pois é tanto um meio para interação usuário-sistema, quanto uma ferramenta que oferece os instrumentos para este processo comunicativo.

Duas são as teorias que dão suporte à IHC, a Engenharia Cognitiva e a Engenharia Semiótica, ambas apresentadas a seguir, de maneira sintetizada.

2.1.1 ENGENHARIA COGNITIVA

A **Engenharia Cognitiva** (NORMAN, 1986), (SOUZA *et al.*, 1999) tem raízes comuns com as áreas de Psicologia Cognitiva, Ciência Cognitiva e Inteligência Artificial que estudam a cognição, isto é, o processo pelo qual o conhecimento é adquirido, e aplicam suas teorias na compreensão das capacidades e limitações dos usuários. Os resultados são de longe mais numerosos do que os de qualquer outra abordagem. 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 usados na interação e realizar experimentos ou previsões com estes modelos.

A idéia básica é que modelos cognitivos que descrevem os processos e estruturas mentais (tais como recordação, interpretação, planejamento e aprendizado) podem indicar para pesquisadores e projetistas quais as propriedades que os modelos de interação devem ter, de forma que a interação possa ser desempenhada mais facilmente pelos usuários. Como estas abordagens adotam uma perspectiva centrada nos aspectos cognitivos do usuário, o *design* feito com este embasamento é chamado de "Projeto de Sistemas Centrado no Usuário" (*User Centered System Design – UCSD*), de acordo com (NORMAN, 1986).

As abordagens cognitivas caracterizam-se principalmente por considerar o processo de interação como um problema de mapear (associar) o conhecimento que se tem a respeito da funcionalidade e da interface de usuário do sistema (o modelo conceitual que o usuário tem do sistema) com o modelo que ele tem da tarefa (o modelo conceitual da tarefa). Os problemas identificados nesta perspectiva envolvem o estudo das atividades mentais necessárias à interação como a recordação de comando e regras e a interpretação de símbolos e resultados, etc. Este mapeamento apenas pode ocorrer no momento em que o usuário tiver adquirido o referido modelo conceitual do sistema (LEITE, 1998). Sob este prisma, o projeto de interfaces inicia-se no modelo mental do projetista, que deve criar uma imagem do sistema que permita ao usuário, durante o processo de uso, desenvolver um modelo mental que retrate as reais intenções do projetista ao desenvolvê-lo.

Dois golfos devem ser atravessados no processo de interação usuário-sistema (NORMAN, 1986). O golfo da execução envolve formulação de metas, determinação de seqüência de ações e atividade física de execução. Já o golfo de avaliação deve ser atravessado através de percepção, interpretação e avaliação do objetivo. A figura 2 relata as etapas necessárias à transposição dos golfos.

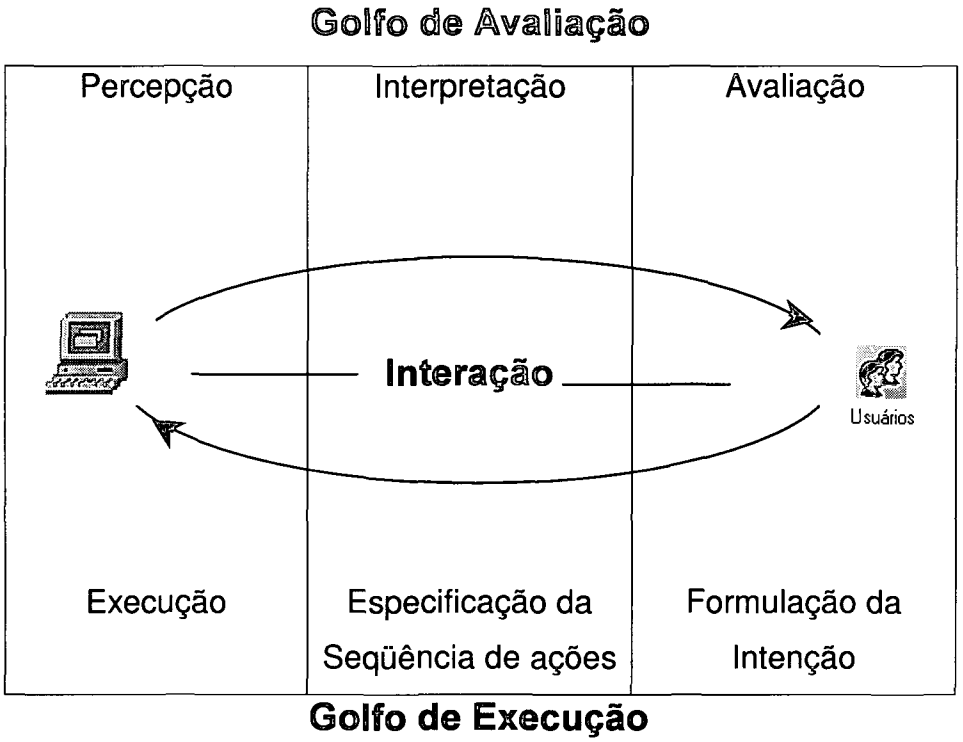


FIGURA 2 - ETAPAS DE AÇÃO NA INTERAÇÃO USUÁRIO-SISTEMA
FONTE – Adaptado de (NORMAN, 1986)

É necessário diminuir os esforços do usuário na execução das etapas de ação aproximando a interface do usuário, retirando deste a necessidade de realizar esforço cognitivo na direção da mesma (cruzar os golfos de execução e avaliação). E isto é tarefa do projetista, que deve diminuir o espaço entre os golfos, fornecendo *feedback*, recursos visuais e de linguagem claros. Com este intuito, dois novos enfoques devem ser minimizados no projeto: a Distância Semântica e a Distância Articulatória.

A Distância Semântica expressa o quanto a interface representa todas as funcionalidades do software, permitindo ao usuário acesso a todo o seu potencial.

No golfo de execução, esta reflete quanto da estrutura requerida é expressa pelo sistema e quanto pelo usuário e será maior quanto maior for o esforço cognitivo realizado pelo usuário na elaboração da solicitação de uma tarefa. No golfo de avaliação, a distância semântica será grande quando o usuário necessitar mapear a saída em expressões compatíveis com a meta a ser atingida.

A Distância Articulatória expressa se todas as operações do *software* estão representadas naturalmente para o usuário e se esta representação é expressa em uma linguagem clara, seja ela textual ou visual. Em ambos os golfos, a distância articulatória pode ser minimizada por meio de qualquer relação não arbitrária entre a semântica e a forma, como denota a figura 3.

A minimização das distâncias semântica e articulatória constitui-se em requisito imposto pela Engenharia Cognitiva ao projeto de interfaces.

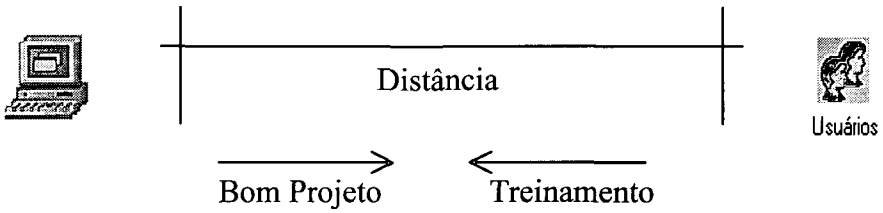


FIGURA 3 - DISTÂNCIA ENTRE O USUÁRIO E O SISTEMA
FONTE - (GARCÍA, 2000)

2.1.2 ENGENHARIA SEMIÓTICA

A **Engenharia Semiótica** tratada em (SOUZA, 1993), (SOUZA *et al.*, 1998), (PRATES *et al.*, 1998), (LEITE, 1998), (SOUZA *et al.*, 1999) traz, para o âmbito do *design* de interfaces, conceitos e resultados da Semiótica, disciplina que estuda os signos, a comunicação e os processos necessários para se criar e compreender os próprios signos.

Para a Engenharia Semiótica, um projetista, por meio da interface criada por ele, comunica-se com o usuário final. Como em todo processo de comunicação, há formulação de perguntas e elaboração de respostas e este processo é concretizado pela interação com o sistema. Assim, a interface é vista, sob esta ótica, como uma

mensagem enviada pelo projetista ao usuário. Nessa hipótese, o foco do *design* de interface deve estar em facilitar essa comunicação, por hipótese unidirecional. Como a interface, por sua vez, transmite mensagem, o processo é de metacomunicação.

Para a Engenharia Semiótica a comunicação é a atividade que coloca usuários e *designers* juntos e a linguagem usada neste processo, deve ser, de conhecimento mútuo. Mas, para isso, o modelo conceitual da aplicação do ponto de vista do projetista e o modelo da aplicação percebido pelo usuário, embora distintos, devem ter consistência entre si. Sistemas são observados como máquinas que produzem e processam elementos de signos, enquanto que os usuários são participantes dos processos que definem um signo.

Segundo (SOUZA, *et al.*, 1999), a Engenharia Semiótica ressalta ainda que a presença do *designer* no cenário comunicativo deve ser explicitada e tornada perceptível para os usuários para que eles tenham maior chance de entender as decisões tomadas e a aplicação com que estão interagindo, sendo assim capazes de fazer um uso mais criativo e eficiente desta aplicação.

Conforme (NADIN, 1988 *apud* LEITE, 1998) a premissa para se considerar a interface de usuário a partir de um ponto-de-vista semiótico é a de que ela representa um sistema de signos complexo, a linguagem, que representa a sua aplicação.

Pela vertente da Engenharia Semiótica, o *design* de interfaces não pode ser considerado a partir de uma perspectiva centrada-no-sistema ou centrada-no-usuário apenas, embora, deva ser ressaltada a importância do contexto e situação de uso na qual se encontram os usuários como peças fundamentais para a interação.

Ao trazer o *designer* para dentro do foco, a Engenharia Semiótica evidencia a sua presença e permite ao usuário entender que todo sistema é uma solução potencial de um *designer* (ou de uma equipe de *design*). Assim, o usuário, ao ter problemas de interação com a aplicação, pode tentar entender o que o *designer* pretendia e acertar o seu modelo mental da aplicação, aproximando-o cada vez mais daquele do *designer*. Fazendo isto, o usuário é capaz de alcançar um melhor

entendimento das motivações e decisões tomadas pelo *designer*, e assim, usar a aplicação de forma mais eficiente (SOUZA *et al.*, 1999).

De acordo com SOUZA (1993), a Engenharia Semiótica vem complementar a Engenharia Cognitiva, oferecendo mais recursos para que se possa estreitar a distância entre a aplicação e o usuário, por meio de um desenvolvimento sistêmico, com enfoque na qualidade de apresentação, garantindo maior comunicabilidade à interface. Desta forma, os formalismos da Engenharia Semiótica contribuem no processo de facilitar o atendimento aos requisitos da Engenharia Cognitiva.

Segue, a descrição de dois formalismos de orientação para a construção de interfaces, a Linguagem de Especificação da Mensagem do *Designer*, que auxilia na definição da mensagem sobre o modelo de usabilidade e a STAG *Semiotic Task-Action Grammar*, que objetiva identificar as tarefas a serem disponibilizadas aos usuários e representá-las de forma categorizada.

2.1.2.1 LEMD – LINGUAGEM DE ESPECIFICAÇÃO DA MENSAGEM DO *DESIGNER*

A interface de usuário é a parte do artefato de *software* com a qual o usuário entra em contato - física, perceptiva, e cognitivamente - na realização de tarefas no seu domínio de atividades (MORAN, 1981). Portanto, uma interface usuário-sistema de qualidade deve auxiliar ao usuário na obtenção de um máximo de produtividade de um dado software, minimizando o esforço despendido na realização desta tarefa (PRADO, 1998). O que se busca é aumentar a eficiência do *software*, por meio da comunicabilidade e a usabilidade do sistema.

Segundo (LEITE, 1998), usabilidade deve ser vista como a qualidade que ao mesmo tempo satisfaz as necessidades de usuário, se acopla às capacidades e conhecimentos de usuários, considera o impacto da tecnologia no contexto de trabalho e integra o usuário no seu contexto de trabalho. O desafio para a usabilidade requer o *design* de equipamentos integrados ao ambiente de trabalho de maneira a aumentar as capacidades (principalmente as intelectuais) de usuários, considerado-os como pessoas inteligentes dotadas de capacidade de compreensão,

aprendizado, interpretação e expressão, ao invés de restringi-los, considerando-os como "idiotas" destinados à execução de trabalhos mecânicos. Ou seja, é necessário ver a aplicação como fornecedora ao usuário de mecanismos para que este compreenda os propósitos de seu *design*, independentemente do seu grau de experiência face à tecnologia.

A comunicabilidade é fator não menos importante, que diz respeito ao processo que ocorre num contexto onde dois ou mais agentes, no papel de fonte e receptor, são interconectados por um canal (*medium*) para veicular uma mensagem que denota um conteúdo. Engloba também fatores como facilidade de aprendizado, facilidade de uso, satisfação do usuário, flexibilidade e produtividade.

Na perspectiva de metacomunicação da Engenharia Semiótica a interface deve desempenhar não apenas funções de entrada e saída, mas, também, revelar a funcionalidade do sistema e, principalmente, os modelos de interação, adquirindo uma função metacomunicativa. Na função metacomunicativa a interface apresenta o MUA (Modelo de Usabilidade da Aplicação), a funcionalidade e o sistema de interação por meio dos quais o usuário pode interagir com o sistema. Esta função pode ser explícita ou implícita. A função explícita ou tutorial ocorre quando uma mensagem da interface se refere diretamente ao sistema de interação. A função metacomunicativa implícita ocorre quando o *designer* se refere a elementos do modelo de usabilidade da aplicação por meio de mensagens da própria interface, isto é, dos seus objetos (LEITE, 1998).

Na abordagem da Engenharia Semiótica a mensagem do *designer* estende o modelo de usabilidade oferecendo o apoio à realização de tarefas por meio do envio de mensagens de metacomunicação direta e indiretamente via interface. Com estes objetivos em mente e visando um mapeamento mais direto com os tipos expressivos, os tipos semânticos devem permitir a descrição do modelo de usabilidade sob a forma de mensagens a serem enviadas ao usuário, compreendendo (LEITE, 1998), (PRADO *et al.*, 2000):

- 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 seqüências *(sequence)*, repetição *(repeat)*, agrupamento *(join)*, combinação *(combine)* e seleção *(select)*;
- Mensagens sobre interações básicas: indicam ao usuário a interação a ser desempenhada. As interações básicas previstas pela linguagem são acionar *(activate)*, fornecer informação *(enter)* e selecionar informação *(select)*;
- Mensagens de meta-comunicação de assistência a tarefas: auxiliam o usuário a realizar tarefas compostas por mais de um comando, apresentando a tarefa decomposta em passos *(Task-Message)*;
- Mensagens de meta-comunicação para apresentação e controle da leitura da mensagem: comunicam como o usuário deve ler a própria mensagem do *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 por meio do elemento *View* da linguagem.

Todas as mensagens recém descritas, quando integradas, compõem a mensagem global do *designer*.

A LEMD permite uma melhor formulação do modelo de usabilidade a ser comunicado, fornece recursos ao projetista para a estruturação da mensagem que revelam as distinções conceituais do modelo de usabilidade e suas estruturas possibilitam, ainda, obter um mapeamento entre estas e os tipos expressivos.

A especificação da mensagem do *designer* inclui a descrição das funções aplicadas, dos signos do domínio, dos comandos e das visualizações, bem como de todas as mensagens que o *designer* envia para o usuário sem a preocupação de com quais *widgets* e configurações espaciais e temporais elas serão expressas. O *design* da interface de usuário deve estar associado semanticamente à especificação da funcionalidade e do modelo de interação do sistema. Esta associação semântica é facilitada por meio da especificação abstrata da mensagem usando a LEMD que permite descrever os componentes do objeto da mensagem. Signos do domínio, funções aplicadas, visualizações e, principalmente, comandos de funções devem poder ser descritos na LEMD para serem representados por *widgets* (LEITE, 1998).

Considera-se que a LEMD:

- é um modelo para o processo de *design* como produção de signos;
- é um meta-modelo para o Modelo de Usabilidade como objeto da Mensagem do *Designer* que descreve quais componentes funcionais e de interação que o usuário precisa aprender;
- descreve os seus componentes e como eles podem ser utilizados para desempenhar as funções de acionamento, revelação e metacomunicação;
- é um sistema semiótico de apoio ao *design* de interfaces de usuário que oferece um formalismo (uma linguagem de especificação da mensagem do *designer*) para a elaboração do modelo de usabilidade a ser expresso por meio da interface cujas sentenças podem ser mapeadas semanticamente por meio de regras de correlação a *widgets* das principais ferramentas de interfaces.

2.1.2.2 STAG (*SEMIOTIC TASK-ACTION GRAMMAR*)

A STAG (MARTINS, 1998), uma evolução da TAG (*Task-Action Grammar*) (PAYNE & GREEN, 1996) surgida como resposta às necessidades específicas do *design* de interfaces gráficas, é uma notação para auxílio à especificação de LVs, que não se propõe a determinar o que deve ser usado, mas a orientar para a sistematização de escolhas expressivas. Incorpora implicitamente alguns princípios básicos da Engenharia Semiótica, de forma que estes princípios sejam automaticamente respeitados, pela simples utilização da mesma (MARTINS, 1998).

A STAG possui quatro componentes: a declaração de tipos e instâncias, o mapeamento entre tipos e instâncias de conteúdo e expressão, um dicionário de tarefas simples (opcional) e um conjunto de regras (uma gramática), para distinguir os mapeamentos. Estes componentes são utilizados dentro de uma estrutura hierárquica, por seções paralelas, permitindo a alternância durante o uso da linguagem (MARTINS, 1998).

O objetivo da notação é registrar de que forma sistema e usuário se expressam a cada iteração e também, auxiliar o projetista a tratar a linguagem de interface de forma sistemática, evidenciando eventuais falhas na linguagem.

Formalismos de especificação do modelo de interação, tais como gramáticas de tarefas-ações, como a STAG, são complementares à LEMD e podem ser integrados a ela.

2.2 PROCESSO DE *DESIGN* DE INTERFACES

Como o processo de *design* deve ter o comprometimento do projetista com os processos cognitivos e comunicativos do usuário, faz-se necessário um planejamento dessa tarefa, visando gerar modelo e documentação formais para a orientação do projetista ao longo do projeto e da implementação.

Souza (SOUZA *et al.*, 1999) afirma que o processo de *design* de interfaces inicia-se com a análise de usuários e tarefas (que constitui a análise de requisitos) e deve ser conduzido num processo cíclico ou iterativo no qual cada passo apresenta

evoluções da etapa anterior. Cada ciclo envolve a especificação da funcionalidade e do modelo de interação, a prototipação de interfaces (que possibilita a interação de acordo como o modelo especificado) e a sua avaliação junto aos usuários. A partir desta avaliação, um novo ciclo de especificação, prototipação e avaliação deve ser realizado, conforme figura 4:

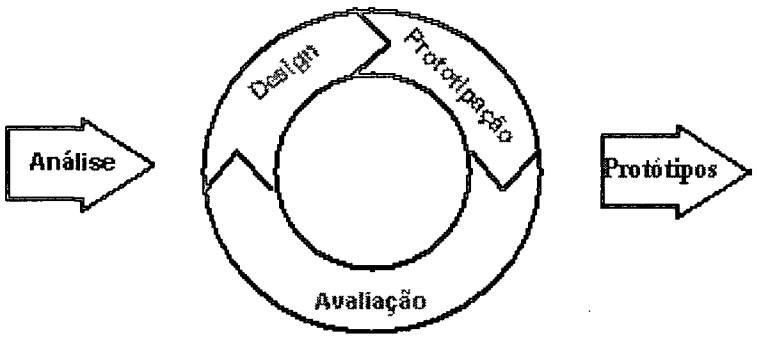


FIGURA 4 - ETAPAS DO PROCESSO DE *DESIGN* DE INTERFACES
FONTE - (LEITE, 1998)

O *design* envolve a concepção e a especificação da interface. A sua prototipação pode ser realizada utilizando-se ferramentas baseadas em *widgets* (dispositivos de interação que são apresentados na tela para serem acionados com o *mouse* ou outro dispositivo equivalente). 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 (LEITE, 1998).

Faz-se necessária uma metodologia formal, sistemática, de construção de linguagens de interface visuais, para garantir a articulação dos símbolos necessária para minimizar o esforço cognitivo do usuário na percepção e no uso dos mesmos na execução da tarefa em questão (aqui representada pela hipótese do trabalho – o processo de elaboração de consultas a banco de dados).

3 MODELO DE DADOS

Segundo (ELMASRI & NAVATHE, 2000) 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.

Os modelos de dados conceituais, ou de alto nível, utilizam conceitos que são próximos da maneira de como o usuário visualiza aquele dado no mundo real, enquanto os modelos físicos, ou de baixo nível, provêem conceitos que descrevem os detalhes de como os dados são armazenados no banco de dados. Estes conceitos são compreendidos somente por pessoal especializado da área de Computação, enquanto os modelos conceituais podem ser entendidos pelos usuários finais dos bancos de dados (RODACKI, 2000).

Um dos principais e mais difundidos modelos de dados é o modelo Entidade-Relacionamento proposto por CHEN (1976), que representa a percepção do mundo real a partir de um conjunto de objetos básicos, definidos como entidades, e seus relacionamentos.

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 (CHEN, 1976).

Um papel é associado a cada participação de uma entidade em um relacionamento. Este papel é caracterizado por seu mínimo e um máximo de cardinalidade, 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.

Contudo, este modelo de dados deixa a desejar quando o incremento do poder semântico do banco de dados é esperado.

O modelo ERC+ apresentado por (SPACCAPIETRA *et al.*, 1995a) e utilizado em (SPACCAPIETRA *et al.*, 1995b), (RODACKI, 2000), (GUEIBER, 2001), estende o modelo entidade relacionamento suportando características do modelo orientado a objetos. O diagrama representando este modelo será apresentado na seqüência.

Este modelo, em um nível conceitual, incorpora relacionamentos genéricos n-ários, estruturas de objetos complexos/compostos e duas linguagem formais, as quais dão suporte ao próprio modelo apoiando a descrição de objeto complexo e sua manipulação, o que resulta em paradigma e metodologia claros baseados nas fundamentações de linguagem de programação favorecendo o desenvolvimento de linguagens e interfaces orientadas ao usuário, pois objetiva uma maior descrição semântica.

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 linhas simples tracejada, se o atributo for opcional. Os atributos sublinhados indicam atributos identificadores. 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.

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 obrigatório, 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.

Duas generalizações são permitidas no modelo ERC+, "*is-a*" e "*may-be-a*".

Uma generalização "is-a" corresponde ao conceito onde se identifica uma entidade genérica que possui as características principais de entidades específicas e é indicada por meio de uma linha contínua com uma seta apontando para a entidade genérica. Já 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 e é representada por uma linha tracejada apontando para a entidade genérica.

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.

Um dos principais problemas das linguagens de manipulação de dados desenvolvidas para trabalhar com modelos objeto-relacional e extensões do modelo entidade relacionamento é não dispor de operadores para trabalhar com todos estes conceitos, especialmente operadores para manipular diretamente relacionamentos genéricos (GUEIBER, 2001).

O modelo ERC+ é complementado com definições formais por meio de uma linguagem de manipulação, com uma álgebra associada, para que seja possível manipular consultas em um banco de dados ERC+.

A álgebra ERC+, definida em (SPACCAPIETRA et al., 1995a) 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+ pode ser satisfeita por uma expressão algébrica apropriada.

Em (GUEIBER, 2001), a Álgebra do Modelo ERC+ é descrita como segue:

A propriedade de fechamento 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, da seguinte forma:
  - seu identificador de objeto (oid[1]) 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 ao modelo comportar objetos complexos os operadores constróem seus tipos de entidades resultantes como objetos complexos. Em particular os operadores *product* (produto), *relationship-join* (junção relacional) e *identity-join* (junção identidade) (usados para transformar diferentes tipos de entidades dentro de um simples objeto) utilizam 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 planares[2]. A ocorrência dos outros operandos são transformadas dentro de um atributo complexo multivalorado.

Esta particularidade da álgebra ERC+ tem duas principais vantagens:

- 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 por meio de ligações *is-a* (é-um) e *may-be-a* (pode-ser-um) é efetuada de duas maneiras.

---

[1]É a abreviatura para o termo em inglês *object identificator* - identificador do objeto. Os identificadores de objetos são únicos, isto é, cada objeto tem um identificador e não há dois objetos com o mesmo identificador. Uma entidade mantém sua identidade independentemente de alterações do valor de suas propriedades (SILBERSCHATZ et al., 1999).

A álgebra oferece um operador, o *identity-join*, que permite ao usuário agrupar dois tipos entidades por meio 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. No anexo A estes operadores são brevemente apresentados.

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

4 EVOLUÇÃO DE LINGUAGENS TEXTUAIS DE CONSULTA PARA LINGUAGENS VISUAIS DE CONSULTA

Após a evolução dos meios de armazenamento de dados, uma das grandes preocupações da área de Banco de Dados está centrada na extração dos dados, ou seja, na melhor forma de recuperação, pelo usuário, dos dados armazenados. A seguir, serão descritas as linguagens de consulta clássicas na literatura, as ferramentas visuais de consulta, e, ainda, o ponto de vista do usuário final da aplicação.

4.1 LINGUAGENS DE CONSULTA

Segundo (DENNENBOUY, 1995) existem duas formas de extração de dados em um banco de dados. A primeira é a intencional, onde se formula uma consulta declarando-se quais propriedades devem ser atingidas pelos objetos e/ou instâncias em uma base de dados para que sejam selecionados e incluídos no sub-esquema da consulta a ser gerado apenas com as entidades, os atributos e os relacionamentos relativos à população relevante à consulta.

A outra forma é extensional, onde se navega através da base de dados até o nível de ocorrências, selecionando os dados que interessam. A navegação implica a seleção de elementos visuais, a captura de elementos adjacentes e a criação de conexões não existentes no diagrama inicial, ou seja, aspectos inerentes à consulta.

(SILBERSCHATZ *et al.*, 1999) classifica as linguagens de consulta em procedurais e não-procedurais.

Em uma linguagem procedural, como a Álgebra Relacional, o usuário deve "ensinar" ao sistema a realização de uma seqüência e operações no banco de dados para obter o resultado desejado. Em uma linguagem não-procedural, como o Cálculo Relacional de Domínio e o Cálculo Relacional de Tuplas, o usuário descreve a informação desejada sem fornecer um procedimento específico para a obtenção dessas informações. Os sistemas de banco de dados comerciais oferecem

linguagem de consulta que incorporam elementos de ambos os enfoques: procedurais e não-procedurais (SILBERSCHATZ *et al.*, 1999).

A Álgebra Relacional, o Cálculo Relacional de Domínio e o Cálculo Relacional de Tuplas são concisos, dando lugar a linguagens de consulta formais muito distantes da linguagem comum dos usuários, o que impõe alto grau de dificuldade à assimilação.

A SQL (*Structured Query Language*) é uma linguagem de definição e manipulação de dados muito poderosa que se estabeleceu como a linguagem padrão para banco de dados relacionais, sendo, porém, de difícil compreensão pelo usuário final, pois adota a forma intencional de expressão de dados. É uma linguagem para desenvolvedores de aplicações.

A estrutura básica de uma expressão SQL consiste de três cláusulas: *select*, *from*, *where*. A cláusula *select* é usada para listar os atributos desejados no resultado de uma consulta. A cláusula *from* lista as relações a serem examinadas na avaliação da expressão. A cláusula *where* consiste em um predicado envolvendo atributos de relações que aparecem na cláusula *from*.

A SQL forma um produto cartesiano das relações indicadas na cláusula *from*, executa uma seleção em álgebra relacional usando o predicado na cláusula *where* e, então, projeta o resultado sobre os atributos da cláusula *select* (SILBERSCHATZ *et al.*, 1999).

A SQL oferece, ainda, uma rica coleção de recursos, que abrange funções agregadas, ordenação de tuplas e outras capacidades não incluídas nas linguagens formais anteriormente citadas, o que a torna rigorosamente mais poderosa.

Em 1977, Zloof lançou a QBE (*Query By Example*) (ZLOOF, 1977), ÖZSOYOGLU & WANG, 1993), (SILBERSCHATZ *et al.*, 1999) que foi a primeira proposta de uma linguagem de consulta visual para banco de dados relacionais.

A QBE é intimamente ligada ao Cálculo Relacional de Domínio (SILBERSCHATZ *et al.*, 1999) pois, ao invés de fornecer um procedimento para obter a resposta desejada, o usuário dá um exemplo do que deseja-se ter e o sistema generaliza este exemplo para computar a resposta da consulta.

QBE baseia-se na exibição do esquema de relações (tabelas "esqueleto") que o usuário quer manipular, com linhas e colunas necessárias à representação das tuplas a serem definidas, por meio das quais o usuário descreve suas consultas preenchendo as colunas apropriadas para indicar os campos de resultado, os valores de seleção, classificação e as condições de união.

A interface QBE pode ser de fácil uso para consultas simples (que envolvam seleção, projeção e união), mas em consultas complexas requer maior entendimento e uso de mecanismos adicionais como caixas de condição e sintaxe.

Seguem, exemplos de consultas expressas em Álgebra Relacional, Cálculo Relacional de Tuplas, Cálculo Relacional de Domínio, SQL, QUEL e QBE, visando mostrar a evolução das linguagens de manipulação de dados e ilustrar os diferentes estilos de interação – sintaxe das linguagens.

Dado o esquema:

empregado (nome_empregado, rua, cidade)

trabalha (nome_empregado, nome_companhia, salário)

companhia (nome_companhia, cidade)

deseja-se:

1-) Encontrar os nomes de todos os empregados que trabalham para a empresa XYZ.

Consulta expressa em Álgebra Relacional:

Π nome_empregado (σnome_companhia = "XYZ Ltda." $^{(\text{trabalha})}$)

Consulta expressa em Cálculo Relacional de Tupla:

$\{ t \mid \exists s \in \text{trabalha} (s[\text{nome\_empregado}] = t[\text{nome\_empregado}] \wedge$

$s[\text{nome\_companhia}] = \text{"XYZ Ltda."})\}$

Cálculo expressa em Relacional de Domínio:

{ <a> | ∃ b, c (<a, b, c> ∈ trabalha ^ b = "XYZ Ltda.")}

Consulta expressa em SQL:

SELECT nome_empregado
FROM trabalha
WHERE nome_companhia ="XYZ Ltda."

Consulta expressa em QBE:

Trabalha	nome_empregado	nome_companhia	salario
	P.	XYZ Ltda.	

Obs.: P. é um comando que indica o que irá ser visualizado em tela.

2-) Obter os nomes de todos os empregados, no banco de dados, que moram na mesma cidade da companhia em que trabalham.

Consulta expressa em Álgebra Relacional:

Π nome_empregado (σ empregado.cidade = companhia.cidade (empregado ⊠ trabalha ⊠ companhia))

Consulta expressa em Cálculo Relacional de Tupla:

{ t | ∃ s ∈ empregado (t[nome_empregado] = s[nome_empregado] ^ ∃ w ∈ trabalha (w[nome_empregado] = s[nome-empregado] ^ ∃ z ∈ companhia (z[nome_companhia] = w[companhia] ^ z[cidade] = s[cidade])))}

Consulta expressa em Cálculo Relacional de Domínio:

{ <a> | ∃ b, c (<a, b, c> ∈ empregado ^ ∃ d, e (<a, d, e> ∈ trabalha ^ ∃ (<d, c> ∈ companhia)))}

Consulta expressa em SQL:

```
SELECT empregado.nome_empregado
FROM empregado INNER JOIN
    (companhia INNER JOIN trabalha ON
        companhia.nome_companhia = trabalha.nome_companhia) ON
        (empregado.nome_empregado = trabalha.nome_empregado) AND
        (empregado.cidade = companhia.cidade)
```

Consulta expressa em QBE:

Empregado	nome_empregado	cidade	rua
	P._x	_z	

Trabalha	nome_empregado	Nome_companhia	salario
	_x	_y	

Companhia	nome_companhia	cidade
	_y	_z

Existem outras linguagens de consulta declarativas menos poderosas que SQL e/ou menos populares no mercado, como QUEL e também outras linguagens como DATALOG para o paradigma de banco de dados lógicos, bem como extensões da álgebra relacional (como é o caso da álgebra do modelo ERC+) e da SQL para contemplar consultas a banco de dados orientados a objetos ou mesmo para efetuar consultas na *Web*.

Mas, a partir da QBE é que projetos começaram a ser desenvolvidos com o intuito de se definir uma linguagem visual de consultas onde o próprio usuário, de maneira simplificada e com ajuda da própria interface, pudesse extrair as informações de acordo com sua necessidade, sem se preocupar com a programação de suas solicitações.

4.2 A INTERFACE-USUÁRIO PARA BANCO DE DADOS

No contexto de interfaces para banco de dados, procura-se atingir a minimização do esforço despendido na realização da tarefa de elaboração de consultas, bem como padronizar este desenvolvimento.

Embora inúmeros esforços tenham sido feitos nesta direção, a integração entre as aplicações e as bases de dados ainda deixam a desejar, conforme mostra a figura 5, onde vê-se que para cada um dos componentes a informação é codificada em formatos independentes, com características redundantes e/ou incompatíveis.

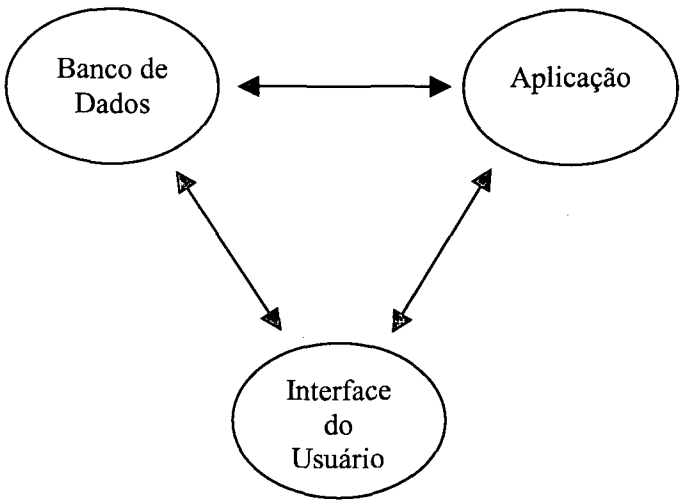


FIGURA 5 - INTERLIGAÇÃO ATUAL
FONTE - (SOWA, 2000)

A necessidade de padronizar maneiras de codificar conhecimento vem sido reconhecida desde os anos 70. A ANSI (*American National Standards Institute*) propôs que todo conhecimento pertinente a um domínio de aplicação fosse coletado em um único esquema conceitual. A figura 6 ilustra um esquema integrado com um esquema conceitual unificado ao centro. Cada círculo é especializado para seu propósito próprio, mas todos fecham em uma aplicação de conhecimento comum, representada no esquema conceitual.

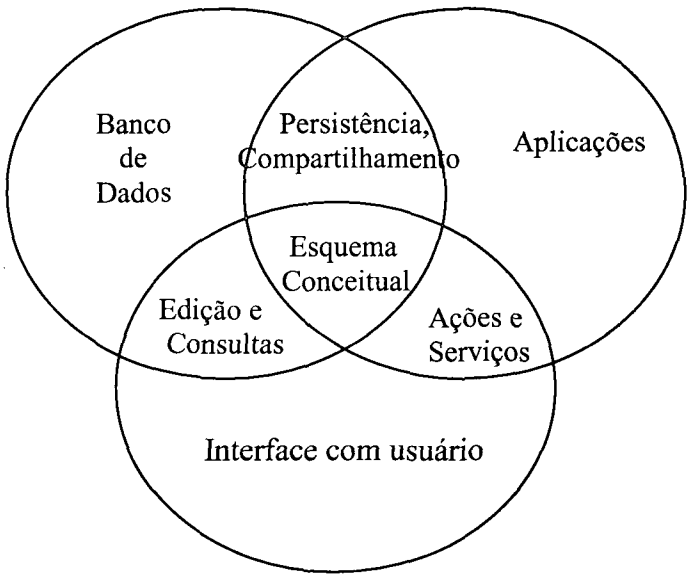


FIGURA 6 - INTEGRAÇÃO IDEAL
FONTE - (SOWA, 2000)

A interface com o usuário acessa o banco de dados para consultas, e este chama os programas de aplicação para realizar as ações e prover os serviços. Então, o banco de dados apoia os programas de aplicação com facilidades para compartilhamento de dados e armazenamento persistente. O esquema conceitual ata os três círculos, fornecendo definições comuns (SOWA, 2000). Isto deve ser revelado pela interface ao seu usuário.

Embora tenha-se buscado esta integração, infelizmente ainda não há uma implementação que englobe todos os aspectos. Apenas implementações parciais são encontradas. A exigência é que se atinja uma integração maior, onde todos os fatores envolvidos estejam em sinergia, muito mais próximos entre si e do usuário.

Os usuários de banco de dados, motivados em parte pela evolução no domínio de aplicação e em parte pela utilização cada vez mais freqüente e consistente de banco de dados, passaram a exigir muito mais quantidade e qualidade de informações, aumentando, desta forma, a necessidade de interfaces adequadas, principalmente, no tocante à manipulação das informações constantes nas bases de dados.

Nos sistemas onde o gerenciamento dos dados normalmente é realizado no nível sintático, é responsabilidade do usuário não somente formular corretamente suas consultas, como também interpretar os resultados advindos destas.

Freqüentemente, a tarefa de formular uma consulta suficientemente precisa para obter a exata informação desejada é complexa, uma vez que a especificação da sintaxe que expresse a consulta semanticamente correta requer um profundo conhecimento do esquema. Para evitar que um usuário não familiar com o esquema perca tempo em um processo de tentativa e erro, (CHEN, 1991 *apud* PORTO, 1997) propõe a utilização de técnicas baseadas em conhecimento, considerando as consultas em um mais alto nível, conceitual, traduzindo-as ao contexto do usuário. Contudo, o usuário não está eximido de abstrair o conhecimento do esquema do BD, porém, como o processo de consulta passa a ser mais expressivo por meio de uma interface visual, a aquisição deste conhecimento, pode ser diluída no processo empírico. Adicionalmente, sempre que uma solicitação é bem sucedida, regras de estruturação começam, de forma condicional, a serem incorporadas no processo mental do usuário, que passa intuitivamente a executar uma operação mentalmente formulada a partir de sua percepção sobre a interface. Isto é o que se busca através de Sistemas Visuais de Consulta. Contudo, manuais, sistemas de auxílio, tutoriais, seções de treinamento são exemplos de ferramentas de apoio também necessárias neste processo de ensino/aprendizado.

De acordo com (PORTO, 1997), uma consulta feita a um BD requer que o usuário possua um aprimorado conhecimento sobre as estruturas de armazenamento de dados (por exemplo, em um BD relacional, deve conhecer entidades, relacionamentos, atributos, entre outros), além da sintaxe e da semântica da linguagem de consulta utilizada, o que não é uma tarefa trivial para um usuário final de aplicação. Cabe à interface preocupar-se mais em como o usuário deseja pensar sobre a base de dados e muito menos com a forma como o SGBD realmente opera, haja visto que com a tecnologia cada vez mais abrangente e disseminada, o desafio agora é explorar toda a sua funcionalidade. A linguagem da interface deve prover uma maneira natural, poderosa e produtiva para o usuário pensar sobre o domínio da aplicação.

4.3 O PROCESSO DE FORMULAÇÃO DE CONSULTAS A BANCO DE DADOS

Uma linguagem de consulta consiste em um conjunto de operadores, definidos formalmente, que possibilitam ao usuário fazer a consultas a uma base de dados, obtendo desta forma resultados, extraídos dos dados armazenados, que são referentes às informações requeridas pelo usuário.

A criação de uma consulta implica a execução de vários passos:

- Selecionar as classes onde se deseja colocar uma restrição;
- Estabelecer as restrições para determinar as instâncias de interesse, selecionando o atributo no qual pretende-se colocar a restrição. Ao selecionar o atributo desejado, as operações que podem ser aplicadas ao atributo são apresentadas. Quando mais de uma restrição precisa ser incluída em uma consulta, surge a necessidade de inclusão de operadores booleanos (*AND* e *OR*) que devem ser selecionados em componente próprio, dependendo da aplicação.
- Selecionar quais atributos serão visualizados nos resultados.

Em seu trabalho, (DENNENBOUY, 1995) propõe um processo de formulação de consultas genérico:

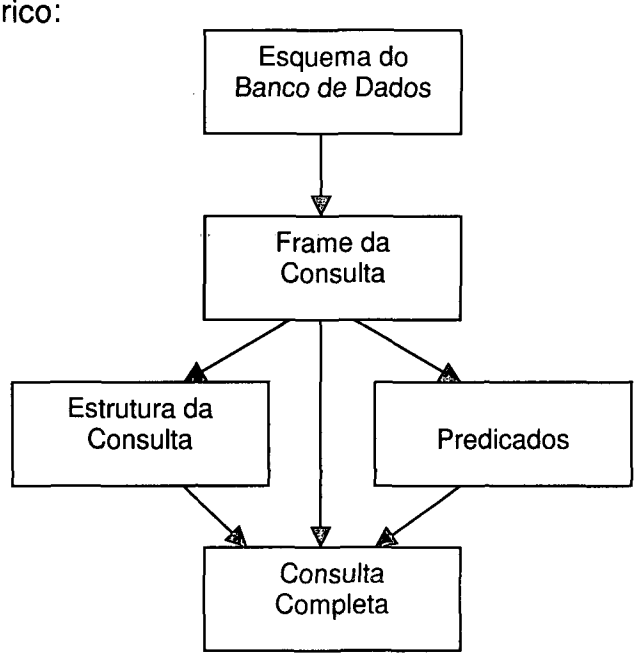


FIGURA 7 - PROCESSO DE FORMULAÇÃO DE CONSULTAS
FONTE - (DENNENBOUY, 1995)

Cinco módulos compõem o processo de formulação de consultas genérico:

- **Esquema do Banco de Dados**: (entrada) por meio do esquema (diagrama) principal do banco de dados são definidas as classes de objetos relevantes para o processamento da consulta, gerando então o *frame* da consulta (subesquema de trabalho).
- ***Frame* da Consulta**: é um grafo acíclico (diagrama) contendo todos os tipos de objetos relevantes para a consulta.
- **Estrutura da Consulta**: é a estrutura do resultado da consulta. São definidos quais dados e de que maneira eles serão apresentados ao usuário como resultado da consulta.
- **Predicados**: é a parte onde são definidos os subconjuntos relevantes expressos por condições (predicados) nos atributos da consulta. Ou seja, compõe-se somente dos objetos que obedecem à condição estipulada pela consulta.
- **Consulta Completa**: é o resultado da consulta, especialmente direcionado ao pedido do usuário (saída).

A partir da figura 7, pode-se observar que durante o processo de formulação da consulta, não existe nenhuma ordem obrigatória entre a definição da estrutura da consulta e a definição dos predicados. Neste caso, como existe uma certa ambigüidade, confusões e erros são comuns na formulação da consulta pelo usuário.

A metodologia subjacente à figura 7 é implementada pela maioria das ferramentas e protótipos existentes, embora não atenda completamente às necessidades dos usuários, visto que este tipo de metodologia faz com que o usuário pense no processo de formulação da consulta como uma composição de sub-consultas, ou, como um refinamento de consultas mais complexas.

4.4 SISTEMAS VISUAIS DE CONSULTAS

No contexto recém apresentado, os sistemas Visuais de Consulta (*Visual Query Systems* - VQSs) descritos em (BATINI *et al.*, 1992), (CATARCI, *et al.*, 1996a), (CATARCI, *et al.*, 1996b) são melhor indicados para atingir a usabilidade e comunicabilidade exigidos na execução da tarefa e na interação com a interface por terem como objetivo aproximar os conceitos do mundo real, percebidos pelo usuário, dos esquemas que descrevem os dados dos sistemas de informação (BRENNAN, 1990). Estes sistemas propõem a idéia de simplicidade de navegação pelo ambiente.

VQSs são sistemas que possuem linguagens de consultas visuais essencialmente baseadas no uso de representações visuais para descrever o domínio de interesse e expressar as solicitações, e podem ser classificados seguindo os eixos:

- **Poder Expressivo**: pode ser definido como a habilidade do sistema extrair informações importantes do banco de dados.
- **Usabilidade**: é especificada em termos de modelos usados para denotar dados, consultas, representações visuais correspondentes e as estratégias fornecidas pelo sistema para formular a consulta.
- **Classes de Usuários**: se referem a que tipo de pessoas o sistema é destinado.

Estas linguagens visuais têm como características principais:

- **O uso de ícones e metáforas visuais**, que atraem mais a atenção do, ao invés de texto;
- **A dispensa ou não de conhecimento prévio** sobre o esquema do banco de dados e da familiarização com a sintaxe de uma linguagem de consulta;
- **A disponibilidade ou não de mecanismos interativos** para suportar o processo típico de formulação de consultas.

A meta das pessoas que trabalham com um VQS é recuperar os dados desejados, o que normalmente é realizado pelas seguintes atividades:

- **Entender a realidade de interesse**: O objetivo desta atividade é a definição precisa do fragmento do esquema envolvido na questão. Corresponde à identificação de elementos úteis à consulta.
- **Formular a consulta**: Esta envolve a manipulação de atributos e operadores disponíveis. A meta da formulação de consultas é expressar formalmente os operandos envolvidos com os operadores relacionados.

Para (BATINI *et al.*, 1992), as principais tarefas que um usuário executa na interação com uma base de dados são:

- Interpretar o conteúdo da base de dados;
- Focar itens de interesse;
- Achar padrões de consulta;
- Inferir conclusões, quando de posse dos resultados das consultas.

Tais tarefas necessitam o uso de técnicas de seleção, navegação, filtragem, controle de *zoom*, *feedback*, entre outras.

A maneira como uma consulta é expressa depende muito da representação visual escolhida. (CATARCI *et al.*, 1996a) afirma que ícones podem melhor representar objetos presentes no banco de dados, ao passo que relacionamentos entre estes podem ser melhor expressos por meio de arestas de um grafo, e coleções de instâncias podem ser facilmente dispostas em um formulário.

Existem VQSs que geralmente restringem a comunicação humano-computador a um único tipo de paradigma de interação. Contudo, o uso integrado de diversos estilos de interação vem se tornando prática na resolução de problemas no uso de *software*. (COHEN, 1991) ressalta que as diversas modalidades devem ser utilizadas de maneira integrada e de forma que elas se complementem, completando a funcionalidade da interface. Logo, os VQSs multimodais, com a presença de vários tipos de paradigmas, cada um com suas características e vantagens, podem melhor ajudar a usuários leigos e experientes na interação com o sistema.

Portanto, uma métrica para uma linguagem visual faz-se muito importante, mas não pode ser definida apenas em termos de poder de expressão da linguagem de consulta base. A avaliação da qualidade de uma linguagem visual deve incluir também duas medidas relativas à flexibilidade:

- Até que ponto, a linguagem permite ao usuário seguir uma seqüência específica na formulação da consulta;
- Quais facilidades são oferecidas para a correção de erros.

Para (DENNENBOUY, 1995), a melhor maneira em que as ferramentas podem implementar estas facilidades é a de fazer correções automaticamente, sempre que os erros aparecem, preservando assim, o trabalho já realizado.

O principal objetivo em se adotar uma representação visual num sistema de consultas é comunicar de forma clara ao usuário as informações referentes ao conteúdo do BD, focando-se em características essenciais e omitindo detalhes desnecessários. As informações ao usuário devem ser apresentadas de forma clara, via interface. Entretanto, as informações devem ser mapeadas em conceitos internos da base de dados, para que desta forma o sistema possa tratar os termos. Para lidar com esta natureza dual, uma representação visual deve se basear num formalismo visual, de forma a poder ser manipulado, compreendido e comunicado por humanos e poder ser manipulado, analisado e mantido pelos computadores.

No entanto, apesar de todas as suas características facilitadoras, as linguagens visuais requerem que o usuário tenha um conhecimento dos conceitos da linguagem e informações referentes ao domínio da aplicação bem como de seus mecanismos de interação.

A apropriação dos conceitos e instrumentos da Engenharia Semiótica possibilita a comunicação entre projetista e usuário necessária à percepção e compreensão dos conceitos do modelo e das informações do domínio constantes no banco de dados.

5 APRESENTAÇÃO E AVALIAÇÃO DE FERRAMENTAS VISUAIS DE CONSULTA

Com o objetivo de exemplificar o potencial e as limitações de algumas interfaces visuais para banco de dados, serão apresentadas e analisadas a seguir, uma ferramenta comercial, o *ORACLE DISCOVERER* e as ferramentas acadêmicas PASTA-3, QDB* (*Query by Diagram*), SUPER e VIQUEN (*Visual Query Environment*), essa última, avaliada com o apoio de um formalismo.

5.1 *ORACLE DISCOVERER*

O *DISCOVERER,* descrito em (VANDIVIER, 2000) e também em (ADAMS, 2002), é uma das ferramentas que compõe o pacote de Suporte a Sistemas de Decisão da ORACLE. Ele habilita profissionais de negócios a localizar e analisar os exatos dados que precisam de forma a tomar melhores decisões sobre seus negócios, sem a necessidade de ter conhecimento de BD.

É uma ferramenta que se propõe a elaborar de forma intuitiva: Consultas *ad hoc*, relatórios, explorações e consultas na *Web* a partir de *data warehouses*, *data marts* ou mesmo banco de dados operacionais relacionais.

A ferramenta já foi projetada para ser operada pelo usuário final, proporcionando ajuda em consultas, possibilidade de análise e construção de relatórios e gráficos, com base em informações previamente armazenadas. É uma ferramenta que possibilita consultas *ad-hoc* e análise multidimensional. O *DISCOVERER* divide-se em 2 módulos: o *Administrator Edition* e o *User Edition*.

O *Administrator Edition* importa automaticamente o modelo de dados relacional e permite renomear e esconder campos, definir perfis de usuários, hierarquias nas dimensões e tabelas resumidas para aumentar o desempenho do sistema. A metainformação, chamada *End User Layer*, fica armazenada no servidor relacional e esconde ao usuário final a complexidade do modelo de dados e linguagem SQL. É o componente responsável pela preparação do ambiente do *Data Warehouse*. Nele o usuário estabelecerá as hierarquias, tabelas sintetizadas, *sorts* e

outros. Esse componente também é responsável por análises orientadas ao desempenho, tais como o uso das tabelas sintetizadas e as consultas mais acessadas.

O *User Edition* permite ao usuário final construir, de uma forma livre, os seus relatórios e gráficos. É o componente de acesso aos usuários finais das pesquisas montadas com o *Discoverer Administration*. Este componente pode ser utilizado para exploração de um *data warehouse*, *data mart* ou um ambiente OLTP. Desenvolvido para a extração e análise de dados, conta com recursos de detalhamento em qualquer nível (*drilling*), paginação (*pivoting*), geração de relatórios *ad-hoc* e gráficos de maneira intuitiva e publicar os resultados na *Web*. O *users* é o componente que interage com o usuário final. Neste componente o usuário poderá criar suas próprias consultas além de analisar consultas prontas ou mesmo alterá-las de acordo com condições de acesso estabelecidas no *administration*.

Entre as facilidades para o usuário final, inclui flexibilidade de consulta e interface amigável.

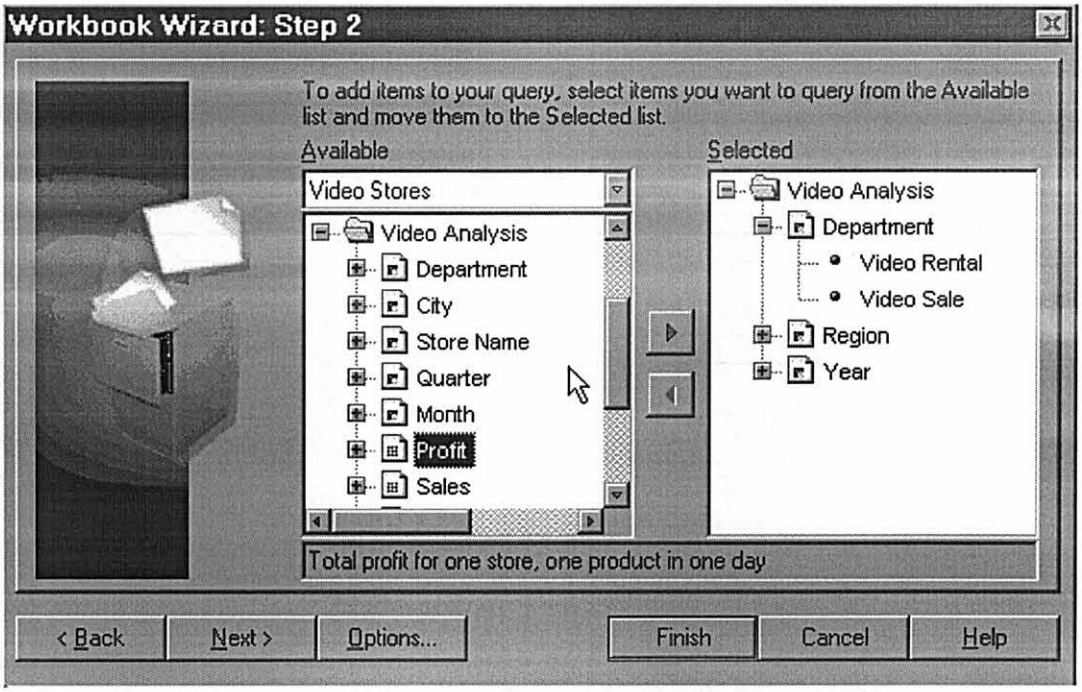


FIGURA 8 - TELA INICIAL DA DEFINIÇÃO DE PREDICADO NO DISCOVERER
FONTE - (ADAMS, 2002)

O DISCOVERER apresenta problemas de interpretação do esquema do banco de dados, que não é visualizado na forma de diagrama, mas sim de maneira hierárquica; muitas opções estão com ativação por menus ocultos, ativados pelo clique com o botão direito *mouse*, o que limita a expressividade dos conteúdos da interface.

5.2 PASTA-3

Pasta-3 (KUNTZ & MELCHERT, 1989), (DENNENBOUY, 1995) é uma interface gráfica para o sistema KB2, um sistema de banco de dados para programação lógica que permite integração entre Prolog e banco de dados relacional em ambiente MS-DOS.

Esta ferramenta é considerada um ambiente cooperativo por fazer manipulação de valores, completar caminhos automaticamente, editar e refazer laços e por possuir grande poder expressivo que permite ao usuário usar condições complexas, executar consultas recursivas, usar variáveis e quantificadores lógicos, definir sub-consultas e fazer interação com ambiente *Prolog*.

Permite consultas em manipulação direta a partir de um ambiente multi-janelas. A edição de consultas é feita por meio do *mouse* (clicando e arrastando) sobre ícones que representam as expressões das consultas.

A interface é constituída de três janelas. A primeira, para formulação de consulta, contém a expressão básica para uma consulta. A segunda janela apresenta a tradução para a base KB2 e a terceira mostra os dados obtidos na consulta.

Depois de criada uma nova área de trabalho, o usuário indica quais as entidades e relacionamentos que estarão envolvidos e opta entre navegar e copiar os itens para a área de trabalho ou fazendo diretamente sua consulta por meio de menus que mostram os objetos envolvidos, escolhidos com o auxílio do *mouse*. Independente da opção escolhida, o diagrama E-R fica sempre disponível, podendo ser acessado por meio de um ícone.

Pasta-3 oferece a possibilidade de conectar entidades e atributos criando novos caminhos não disponíveis no esquema inicial. A expressão de consulta na área de trabalho pode ser modificada a qualquer tempo e, ainda, permite a opção de armazenamento de resultados de consultas anteriores, ou seja, possibilita reuso de consultas, aumentando a produtividade, pois diminui a inserção de erros, a necessidade de memorização e as entradas via teclado.

Neste sistema, consultas complexas podem exigir duplicação de entidades ou uso de ícones que representem quantificadores universais e existenciais. Na realidade, a linguagem permite expressar graficamente uma recursão simples (em uma única entidade e sem condições de igualdade ou diferença) usando a duplicação da entidade, ligada ao mesmo relacionamento por dois diferentes papéis. Recursões complexas podem ser introduzidas por meio da inclusão, em outra janela, de regras Prolog, o que torna o sistema não totalmente visual. O sistema provê o usuário com um conjunto de estratégias e tipos de interação disponíveis nas fases da formulação de consulta.

O sistema pode ser usado pelo usuário novato não somente para fazer consultas simples navegando no diagrama, mas também para adquirir informações sobre o esquema do banco de dados. Os usuários experientes são providos com um grande conjunto de facilidades para realizarem consultas complexas e adquirirem conhecimento profundo sobre o esquema disponível.

PASTA-3 permite recuperação de erros via teclado ou por meio de menus.

O uso extensivo de menus demanda maior ocupação de espaço em tela, aumentando, também, a necessidade de memorização das opções inseridas anteriormente nestes menus. Isso eleva o esforço cognitivo do usuário, que precisa lembrar em qual menu está inserida a opção para execução de uma determinada tarefa.

É impossível, no Pasta-3, a definição direta de vários predicados para uma mesma consulta, o que requer o uso de sub-consultas para solucionar esta limitação. O uso de sub-consultas exige do usuário maior conhecimento na estruturação da expressão, como pode ser visto na figura 9:

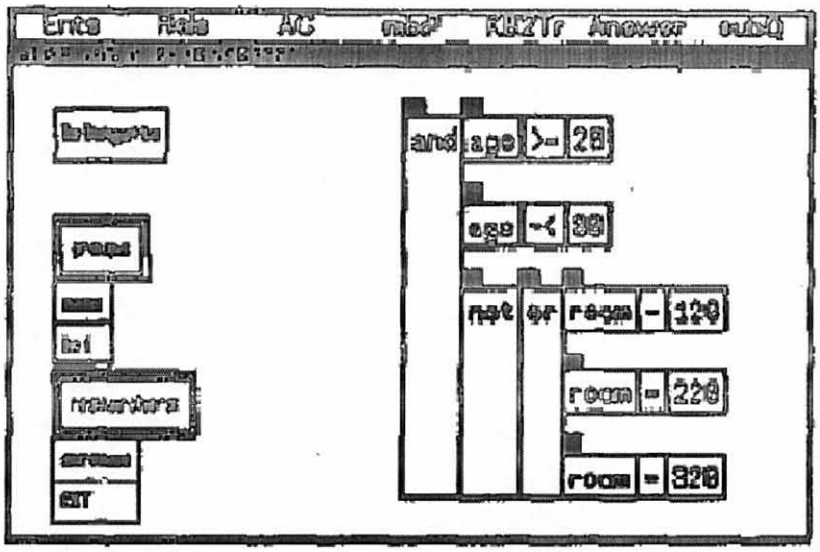


FIGURA 9 - DEFINIÇÃO DE PREDICADO NO AMBIENTE PASTA-3
FONTE - (KUNTZ & MELCHERT, 1989)

5.3 QBD* (*QUERY BY DIAGRAM*)

QBD* (ANGELACCIO *et al.*, 1990a), (ANGELACCIO *et al.*, 1990b), (DENNENBOUY, 1995) é uma linguagem de consulta navegacional que utiliza o diagrama Entidade-Relacionamento (E-R), que permite maior poder de expressividade por meio de consultas recursivas. Suporta interação visual por meio de uma interface gráfica e um conjunto de primitivas para a seleção no esquema do banco de dados e na formulação de consultas.

A estrutura geral da consulta é baseada na seleção de um conceito principal (uma entidade ou um relacionamento), que pode ser visto como um ponto de entrada para uma ou mais sub-consultas. Sub-consultas podem ser combinadas pelo uso dos operadores de união e interseção.

A idéia básica é decompor uma consulta em passos elementares pré-definidos, expressos por um conjunto limitado de operações gráficas, tais como escolha de um ícone ou seleção de um conceito e navegação. Para expressar formalmente a sintaxe e a semântica da linguagem, uma correspondência um-para-

um (mapeamento) entre as operações gráficas e os construtores sintáticos da linguagem de consulta textual foi definida.

Uma vez que o conceito principal tenha sido selecionado, dois diferentes tipos primitivos estarão disponíveis para a navegação no esquema. O primeiro permite que o usuário siga os caminhos existentes no esquema. O segundo é usado na comparação entre dois conceitos para a construção de um novo caminho (relacionamento) no esquema.

Em síntese, a atividade de formulação de consulta é composta de várias fases (seleção, substituição e navegação), cada uma amparada por um grande conjunto de primitivas dispostas ao usuário de acordo com sua necessidade, na forma de um *kit* de ferramentas.

As primitivas gráficas são funções de mapeamento do banco de dados para a estrutura chamada de "esquema de interesse", que é um subconjunto do banco de dados contendo os conceitos necessários à formulação da consulta. As primitivas são agrupadas em três classes, de acordo com o tipo de objeto sobre o qual operam: o grafo (primitivas de seleção), o grafo e a interpretação (primitivas de substituição) e a interpretação (primitivas navegacionais).

O formalismo gráfico usado para representar o modelo conceitual simplifica a execução de atividades típicas envolvidas na formulação de consultas, tais como a extração de um sub-esquema de interesse e a escrita da expressão da consulta desejada.

As operações da Álgebra Relacional podem ser expressas diretamente por meio da seleção por manipulação direta de símbolos do diagrama (entidades ou relacionamentos) e inserindo condições nos seus atributos por meio de um mecanismo de inserção de novas janelas. Consultas recursivas também podem ser expressas pelos mesmos mecanismos, o que elimina completamente a interação textual e a complexidade de algumas consultas, oferecendo-lhe um alto nível de abstração em relação ao formalismo, conforme ilustra a figura 10.

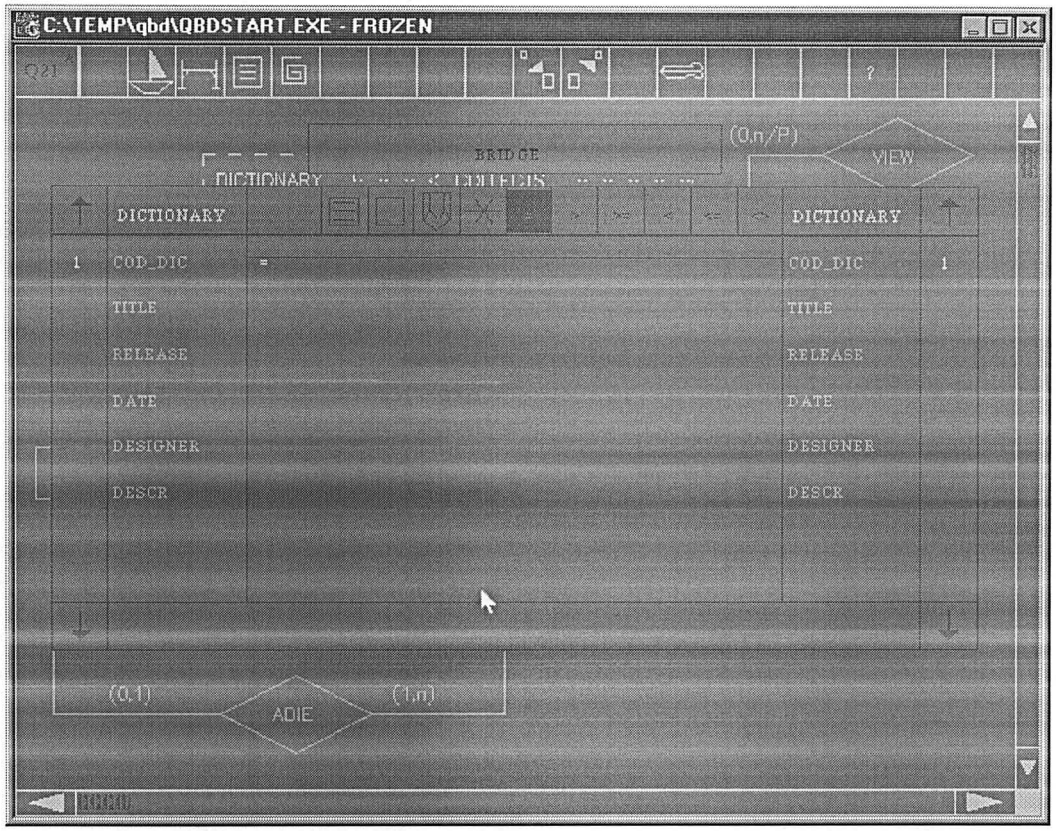


FIGURA 10 - COMPARAÇÃO DE ATRIBUTOS EM QDB*

A interação do usuário com o sistema é reduzida para construir graficamente o grafo da consulta de interesse e construir expressões regulares. A interação não é nem de longe trivial, visto que expressões não são facilmente gerenciadas por usuários novatos, uma vez que estes devem saber criar expressões (inserir atributos, condições de comparação, entre outras).

A interação com o esquema é simples; o usuário não necessita lembrar nomes de tabelas ou atributos, mas somente conhecer a opção para navegação pelo esquema. O tempo necessário para a manipulação direta de objetos na tela é menor que o tempo usado para escrever declarações em uma linguagem textual, o que reduz a taxa de erros, em particular para consultas simples.

O passo mais crítico ao usuário, é onde deve-se utilizar operadores lógicos e/ou valores constantes para inserção de critérios, pois o usuário não está eximido de conhecer a sintaxe da ferramenta. Por exemplo, se a consulta desejada for "Selecione as pessoas que sejam médicas ou advogadas, com um salário superior à

R$4000,00", seria necessário selecionar a tabela, clicar com o botão esquerdo do *mouse* sobre a primitiva de navegação que habilita a inserção de critérios e, sobre o(s) atributo(s) desejado(s) impor os critérios. A expressão referente à primeira parte, como mostra a figura 11, é *="Médico" .OR. *="Advogado" e, no campo salário, a expressão *> 4000 é construída. Nota-se que, o usuário deve conhecer os comandos, sem avaliação prévia à execução de que a sintaxe introduzida está correta.

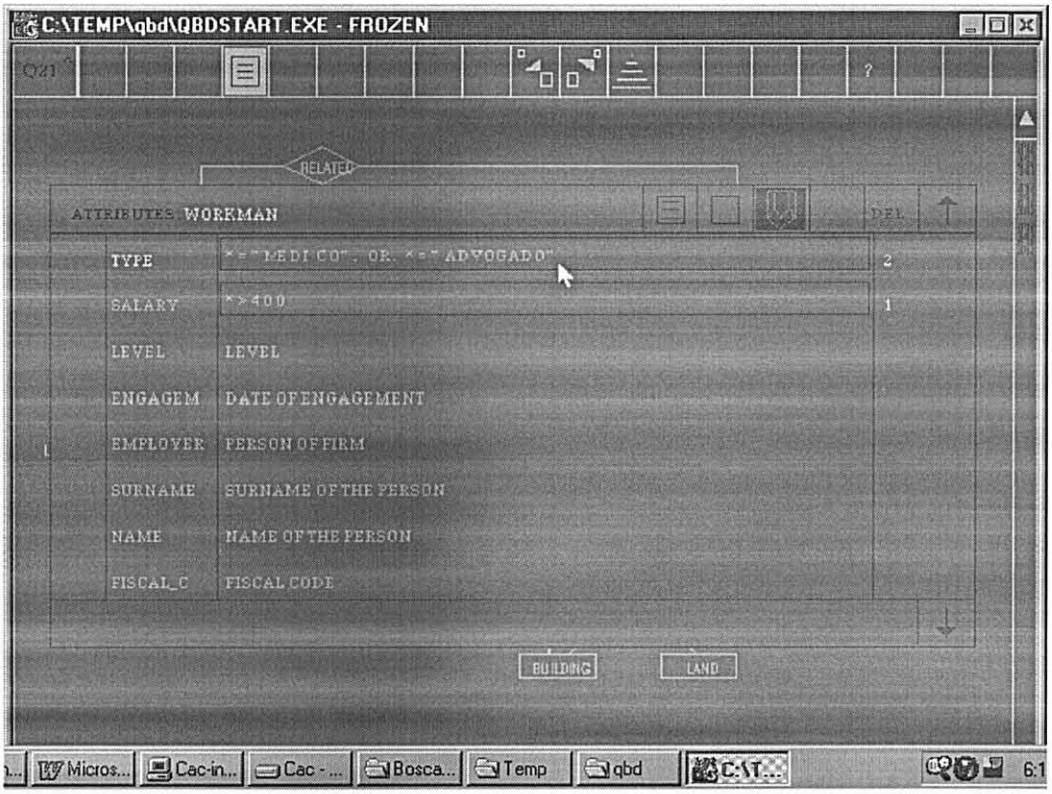


FIGURA 11 - ELABORAÇÃO DE PREDICADOS EM QBD*

O usuário assimila o resultado da consulta pela apresentação gráfica de seus resultados, o que reduz as distâncias semântica e articulatória no golfo de avaliação, pois os comandos e funções estão representados de maneira gráfica e icônica, embora alguns ícones são de difícil interpretação e, são somente esclarecidos no *help* do sistema.

Várias dificuldades podem surgir na transação, decorrentes do desconhecimento prévio do esquema do banco de dados. Entre elas, têm-se: descoberta de informações implícitas, análise muito detalhada do esquema para

entendimento, necessidade de recuperar informações pré-adquiridas, entre outras. Para resolver estes problemas, há um grande conjunto de primitivas gráficas, que exigem, do usuário, uma alta carga de memória para aprendizado dos significados das mesmas.

Como nas propostas anteriores, deve-se partir de um conceito principal. A escolha de um relacionamento errado, por exemplo, levará a uma consulta indesejada. Para solucionar tal problema, o usuário deverá inteirar-se da abstração do modelo E-R (Entidade-Relacionamento), tarefa não trivial a usuários finais da aplicação.

A partir da seleção de objetos no diagrama para a geração do sub-esquema (ou esquema de trabalho), o esquema inicial não é mais utilizado. Portanto, objetos indesejados podem ser eliminados, mas a ausência de um objeto necessário conduzirá à necessidade da re-elaboração total da consulta.

5.4 SUPER

Super (SPACCAPIETRA *et al.*, 1995a), (DENNENBOUY, 1995) é um editor visual projetado para o modelo ERC+, onde o termo "visual" é usado para enfatizar os aspectos de visualização que permitem edição do modelo e formulação de consultas, e não somente representação diagramática do esquema do banco de dados.

Esta ferramenta é baseada na manipulação direta de objetos e funções, com um foco especial em prover ao usuário o máximo de flexibilidade e independência das linguagens formais de banco de dados.

A interface de consulta é consistente com a definição de dados da interface, ou seja, a interação com os usuários é baseada em um conjunto comum de conceitos de modelagem de dados.

As interações na formulação da consulta foram projetadas para obedecer a critérios visuais (tais como legibilidade, intuitividade e entendimento) sem qualquer tentativa para imitar operações de linguagem de manipulação de dados, sequer

havendo menus para estas operações. Também utiliza o modelo ERC+ para interagir com o usuário e não apenas para a exibição do esqueleto completo da base.

Estruturas de dados e operações são baseadas no modelo de dados, e possibilitam manipulação de objetos complexos, relacionamentos explícitos e ligações multi-instâncias.

A construção de uma consulta abrange a definição de:

- **Elementos relevantes:** quais elementos do esquema são relevantes para a consulta dada, que pode incluir uma modificação na estrutura atual destes elementos para satisfazer às necessidades da consulta e para que a interpretação da especificação da consulta não seja ambígua;
- **Condições na população do resultado**. Neste passo o usuário define os predicados que podem ser aplicados às ocorrências do banco de dados, para que somente dados relevantes sejam selecionados;
- **Formatos de saída**. Este passo especifica que itens de dados (atributos) serão incluídos na estrutura de resultado.

Essas três atividades são inerentes ao processo da definição da consulta. Este processo completo seria muito complexo se o usuário tentasse definir todos os passos anteriores em uma única tarefa. Para solucionar este problema, SUPER implementa a separação dos passos usando janelas específicas para cada um.

Super inclui, em particular, características como flexibilidade, reusabilidade e *backtracking*. A flexibilidade permite ao usuário a definição de suas próprias estratégias. Mas as ações não são atômicas, havendo a possibilidade de iniciar uma ação e mover-se para outra sem a necessidade de completar previamente a primeira. Como reusabilidade, entende-se que os usuários podem reutilizar definições de consultas já conhecidas pelo sistema. Por *backtracking* entende-se a possibilidade de o usuário a qualquer tempo poder desfazer ações errôneas e voltar a estados anteriores.

O navegador SUPER implementa dois modos de visualização de dados: um baseado em formulários e outro gráfico. No modo baseado em formulário, a ocorrência ativa é mostrada em um objeto formulário, contendo 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. No modo gráfico eles são apresentados como diagramas E-R mostrando as ocorrências correntes (que estão sendo examinadas), mas realizadas a partir de uma abordagem de manipulação direta.

A possibilidade de o usuário desfazer (*undo*) parte de sua interação e a oferta de diferentes níveis de interação adequados ao diferente grau de conhecimento da ferramenta pelo usuário dão maior flexibilidade à ferramenta.

Na visualização baseada em formulários, o usuário encontrará maior dificuldade na interpretação do resultado da consulta, devido à possibilidade da existência de atributos complexos e multi-valorados que necessitam ser expandidos para mostrar os dados ocultos. Também não se pode inspecionar várias populações ao mesmo tempo, pelo grande número de janelas abertas durante a interação. Desta forma, o usuário necessita aumentar seu esforço cognitivo, o que aumenta também a distância semântica, uma vez que este deve observar que existem campos a serem expandidos e conhecer os conceitos específicos de banco de dados apresentados durante a interação, tais como generalização e especialização.

Conforme pode ser observado na figura 12, o estabelecimento das condições não é uma tarefa trivial ao usuário, que desconhece o formalismo lógico.

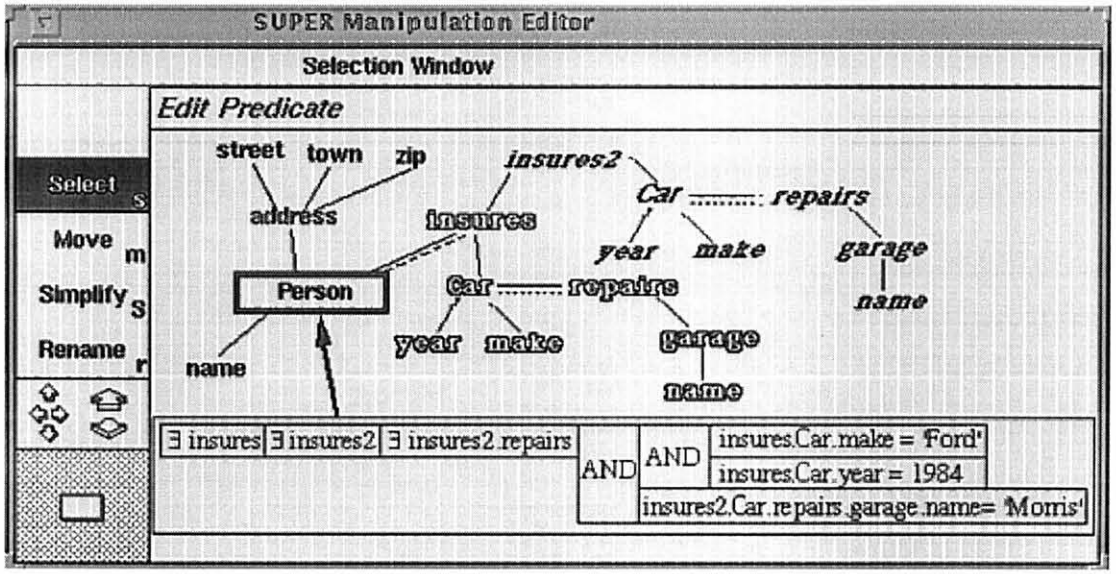


FIGURA 12 - DEFINIÇÃO DE PREDICADO NO AMBIENTE SUPER
FONTE - (SPACCAPIETRA et al., 1995b)

O SUPER fornece suporte completo ao modelo ERC+, mas o faz por meio da utilização de uma máquina abstrata ERC+ para a linguagem de consulta. Assim, a lacuna existente entre o legado armazenado em bases relacionais e o modelo conceitual de objetos permanece.

5.5 VIQUEN (*VISUAL QUERY ENVIRONMENT*)

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 (2001) com o objetivo de atingir a categoria de usuários comuns, que buscam freqüentemente informações armazenadas em sistemas de banco de dados, mas desconhecem as sintaxes de linguagens de consulta, construídas tipicamente para uso por especialistas.

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 e traduz para o modelo ERC+. Esta tradução está baseada exclusivamente nas informações do esquema relacional, mantida no dicionário de dados pelo SGBD relacional, tais como chaves e restrições de unicidade, sobre as quais são aplicadas regras de refinamento para que se atinja maior semântica. Os operadores da linguagem de manipulação de dados do modelo ERC+ são mapeados para SQL, preservando seu poder de expressão e, por outro lado, permitindo acesso ao vasto volume de informações relacionais existente. Apesar de o VIQUEN ter sido construído com o objetivo de atender às necessidades dos usuários não-especialistas, o seu desenvolvimento não tinha a interface-usuário como uma das suas preocupações principais.

O protótipo VIQUEN, conforme descrito em (GUEIBER, 2001), apresenta as seguintes etapas ao 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 que foi selecionado pelo usuário, é traduzido automaticamente pela aplicação para um esquema sob o modelo ERC+.

As tarefas subseqüentes do processo de montar uma consulta neste ambiente (bem como suas sub-divisões), são identificadas na tabela a seguir:

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

TABELA 1 - ETAPAS DO PROCESSO DE CONSULTA DO VIQUEN

A partir do esquema recuperado, o usuário pode selecionar, por meio do *mouse*, os objetos desejados na formação da consulta. Esta seleção deve incluir tanto os objetos desejados na resposta como aqueles necessários na formação do predicado, conforme mostra a figura 13:

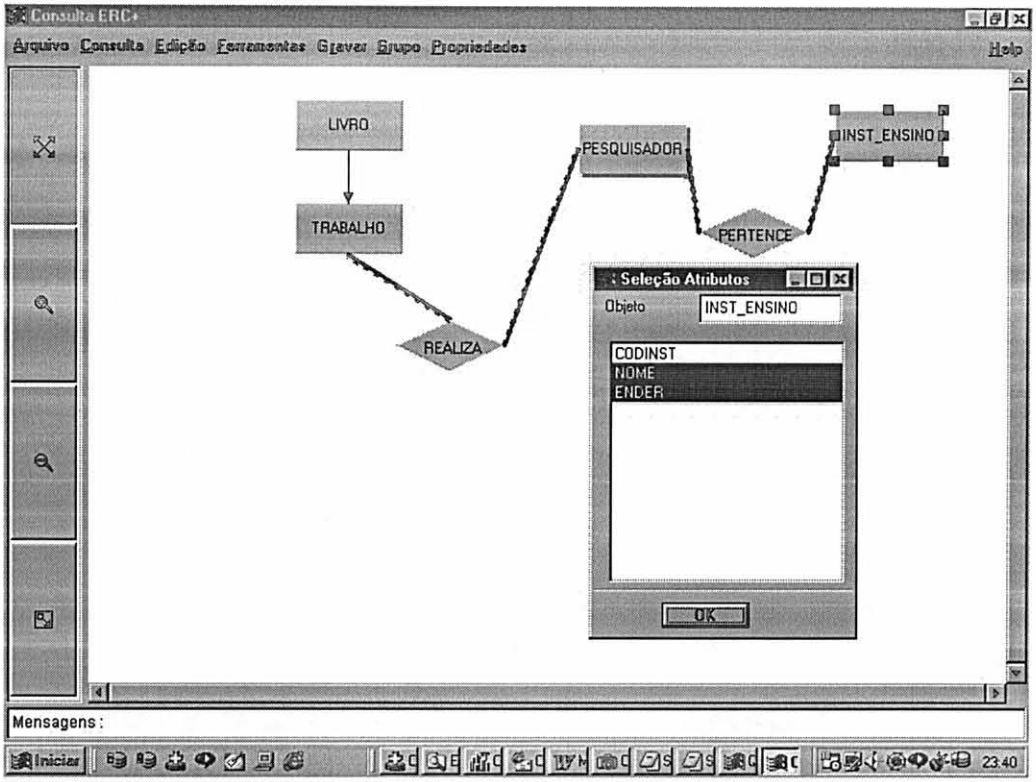


FIGURA 13 - SELEÇÃO DE ATRIBUTOS NO *VIQUEN*
FONTE - (GUEIBER, 2001)

Para submeter uma consulta o usuário deve determinar, também, qual visão deseja, escolhendo uma raiz para a consulta. A seleção da raiz da consulta é uma etapa bastante importante na seleção do esquema, pois é esta escolha que irá determinar a hierarquia dos objetos participantes da consulta a ser realizada. Em alguns casos, é preciso que o usuário elimine eventuais ciclos que haja entre os objetos da consulta. Por último, a fim de complementar a consulta, o usuário pode adicionar sub-esquemas a ela, que devem ser unidos ao sub-esquema inicial, eliminando objetos comuns a ambos.

A fase de 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 das constantes, para construir expressões booleanas sobre os atributos atômicos, conforme figura 14:

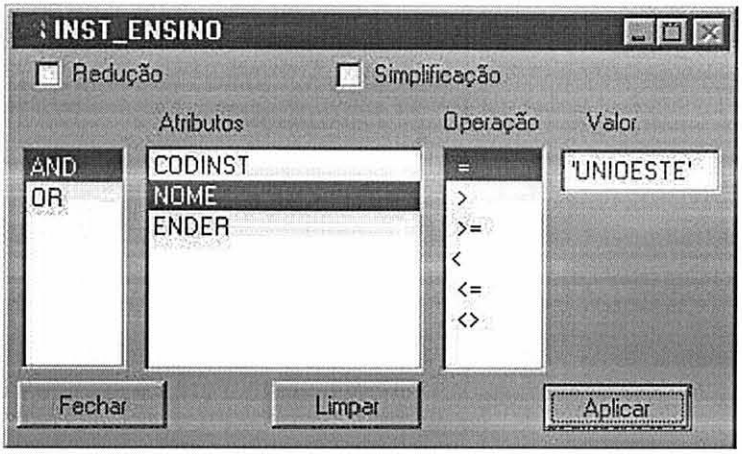


FIGURA 14 - ESTABELECIMENTO DE PREDICADO NO VIQUEN
FONTE - (GUEIBER, 2001)

A etapa de construção do predicado exige, para recebimento eficiente da mensagem, auxílio de fontes externas ao modelo de usabilidade, tais como *helps*, manuais, treinamento ou suporte técnico, pois o usuário enfrentará a necessidade de manipular operadores lógicos, construindo expressões válidas no contexto da consulta em elaboração (o que se constitui numa tarefa de nível de complexidade relevante) e ainda, de optar por uma operação de redução ou simplificação (específicas da Álgebra ERC+). No VIQUEN, não fica claro como estas operações funcionam e o que exatamente elas representam. Desta forma, da interação com o sistema começam a surgir dúvidas quanto ao que escolher e quais as implicações desta escolha, comprometendo o processo de comunicação *designer*-usuário. Ainda nesta etapa, a parte de apresentação em SQL do predicado construído tem papel fundamental no contexto de engenharia reversa. No entanto, esta linguagem possui expressividade mínima para o usuário, por não ser uma linguagem comum em seu cotidiano.

No VIQUEN, uma vez especificado o predicado, são apresentados ao usuário, em forma de árvore, em uma nova janela, os predicados estabelecidos por ele. Esta representação mostra a hierarquia estabelecida pelo usuário quando selecionou a raiz no esquema de consulta. Também é apresentada, na mesma janela, a instrução

SQL que foi automaticamente gerada pela aplicação. A última etapa da aplicação é a visualização da resposta, onde os objetos selecionados durante a consulta permanecem igualmente vistos como objetos. Esta representação, em estrutura não-planar mas de forma hierárquica a partir da raiz previamente selecionada, possibilita que o usuário expanda as instâncias da resposta.

Segue, a descrição em LEMD do VIQUEN no tocante à etapa de construção do predicado (BOSCARIOLI *et al.*, 2001), seguindo a utilização de avaliação proposta por (PRADO & BARANAUSKAS, 2000):

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

Command-Message Predicado ERC+
Combine {
     Command-Message Seleciona Objeto for Application-function
Seleciona Objeto
          Repeat {
               Combine {
                    Select Information-of <Objeto do Esquema >
                    Sequence {
                         Activate Start Application-function
                    Consulta
                         Activate Start Application-function
                    Predicado
                         }
                    }
               }
```

```
Command-Message Complementação de Predicado for
Application-function Complementação de Predicado
     Join {
          Sequence {
               Join {
                    Activate Start Application-function
               "check"
                    Select Information-of "Redução
               Simplificação"
                    }
               Select Set-of (AND OR)
               Select Information-of <Seleção de
          Atributos>
               Select Set-of (= > < >= <= <>)
               Enter Information-of <Valor do Atributo>
               }
          Selection {
               Activate Start Application-function
               Complementação de Predicado
               Activate Stop Application-function
               Seleciona Objeto
               Activate Suspend Application-function
               Complementação de Predicado
                    }
        }
```

O fato de o usuário ter que selecionar atributos, impor raiz, retirar ciclos, entre outras ações, para depois complementar predicados não lhe é revelado.

A análise do protótipo VIQUEN pela LEMD revelou alguns problemas, específicos em cada sub-etapa. Em particular, a etapa de construção de predicados apresenta várias rupturas na metacomunicação da interface, como quando a interface sugere que se opte por uma operação de redução e/ou simplificação (operações específicas da Álgebra ERC+). Não fica claro se ambas as opções podem ser marcadas, como funcionam e o que exatamente elas representam. Faltam informações e, desta forma, na interação com o sistema começam a surgir dúvidas quanto ao que escolher e quais as implicações da escolha, comprometendo o processo de comunicação *designer*-usuário. A fase de aplicação de operadores lógicos (*AND* e/ou *OR*) também não deixa claro o que está sendo feito. As operações e valores selecionados para cada atributo apenas ficam visíveis com a

seleção do mesmo (por meio de clique com o botão esquerdo do *mouse*). Ainda, o botão Limpar não revela se apenas o último critério será excluído ou se todos o serão.

O desenvolvimento de ferramentas que ofereçam ao usuário leigo uma simples compreensão do conteúdo do banco de dados e uma extração simplificada de informações tem demandado esforços da comunidade acadêmica das áreas de Banco de Dados e IHC. Vários outros projetos estão em andamento, para as mais variadas aplicações de banco de dados, tais como *GeoVisual* Interface (FERNANDES & SALGADO, 2000) que provê uma interface de consulta visual para um sistema de informações geográficas e o *Mail-by-Example* (BECKER & CARDOSO, 2000) que destina-se à interface visual de consulta para gerenciamento de *e-mails*, baseado em QBE.

6 INTERFACE PARA ELABORAÇÃO DE CONSULTAS A BANCO DE DADOS

As análises das ferramentas citadas na seção anterior *ORACLE DISCOVERER*, PASTA-3, QDB* (*Query by Diagram*), SUPER e a avaliação do protótipo VIQUEN, base para este trabalho, à luz da Engenharia Semiótica mostraram problemas e possíveis soluções, na tarefa de elaboração de predicados para consultas a banco de dados. Foram assim identificadas algumas situações facilitadoras para o usuário e outras onde a carga cognitiva exigida é muito alta, que representam obstáculos no processo de elaboração de uma consulta. Com base neste levantamento de problemas e soluções, foi desenvolvido o protótipo de interface para o VIQUEN.

6.1 CARACTERIZAÇÃO DO AMBIENTE

São várias as etapas a serem seguidas pelo usuário para realizar uma consulta. A primeira delas é a etapa de seleção de esquema e geração de um sub-esquema de trabalho. Nesta etapa, o usuário adquire uma familiaridade em relação ao esquema visual, que é entrada para a etapa abordada neste trabalho, conforme ilustra a figura 15.

Esta etapa de seleção de esquema e geração de um sub-esquema de trabalho é tratada detalhadamente em (BASSI, 2002) e a apresentação de resultados em (RANTHUM, 2002) sendo, portanto, a fase de elaboração do predicado, ou construção da consulta propriamente dita, o objeto de estudo deste trabalho.

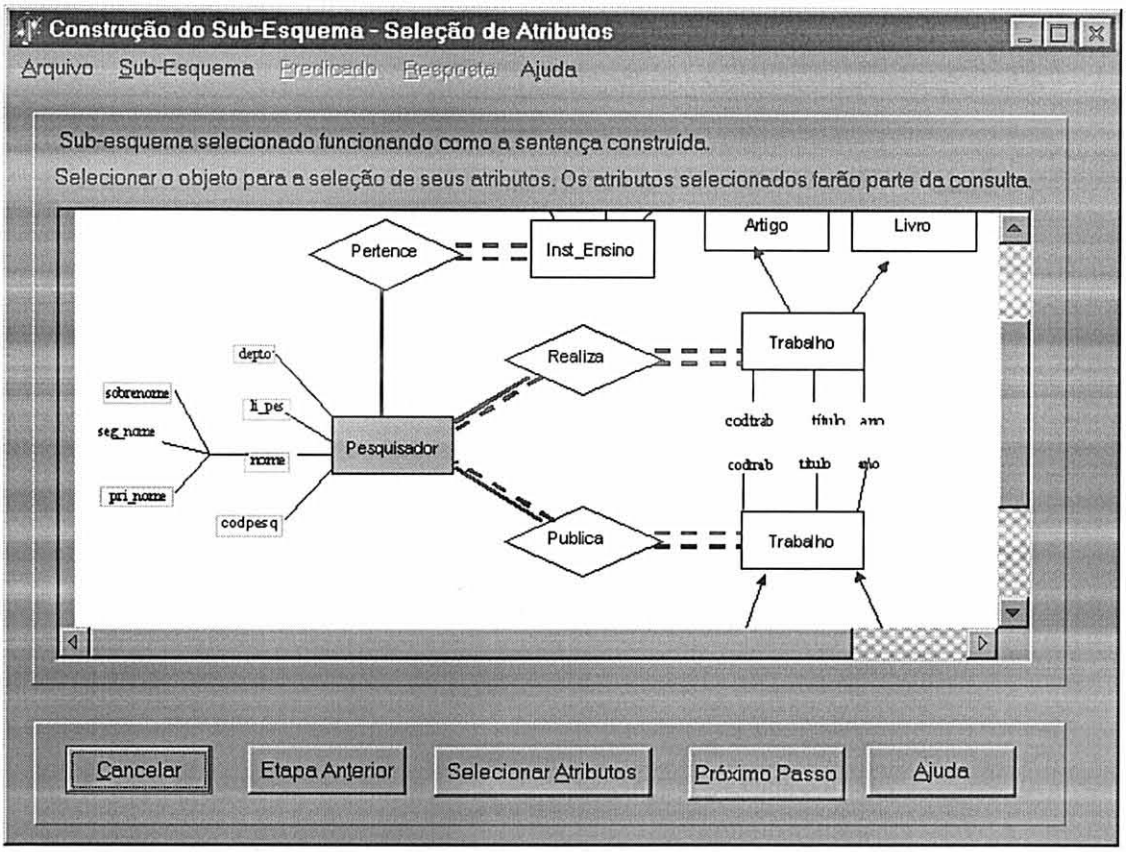


FIGURA 15 - TELA COM UM SUB-ESQUEMA DE TRABALHO GERADO
Fonte: (BASSI, 2001)

6.2 PERFIL DO USUÁRIO

O usuário de hipótese possui experiência em interação com interfaces gráficas - *GUIs* (*Graphical User Interface*) e tem conhecimento do estilo *WIMP* (*Windows, Ícons, Menus e Pointing Devices*). Considera-se também que o mesmo conheça o domínio da aplicação, sabendo exatamente o que deseja como resultado de uma consulta construída. No entanto, o modelo de banco de dados (esquema) não lhe é familiar, o que dificulta a interação em um ambiente visual de acesso a uma base de dados.

6.3 O DOMÍNIO DA APLICAÇÃO

O domínio tomado como hipótese de trabalho foi o de um Sistema para Controle de Publicações da Associação de Professores Universitários do Estado do Paraná, adaptado de (RODACKI, 2000) e modelado sob diagrama ERC+ mostrado na figura 16:

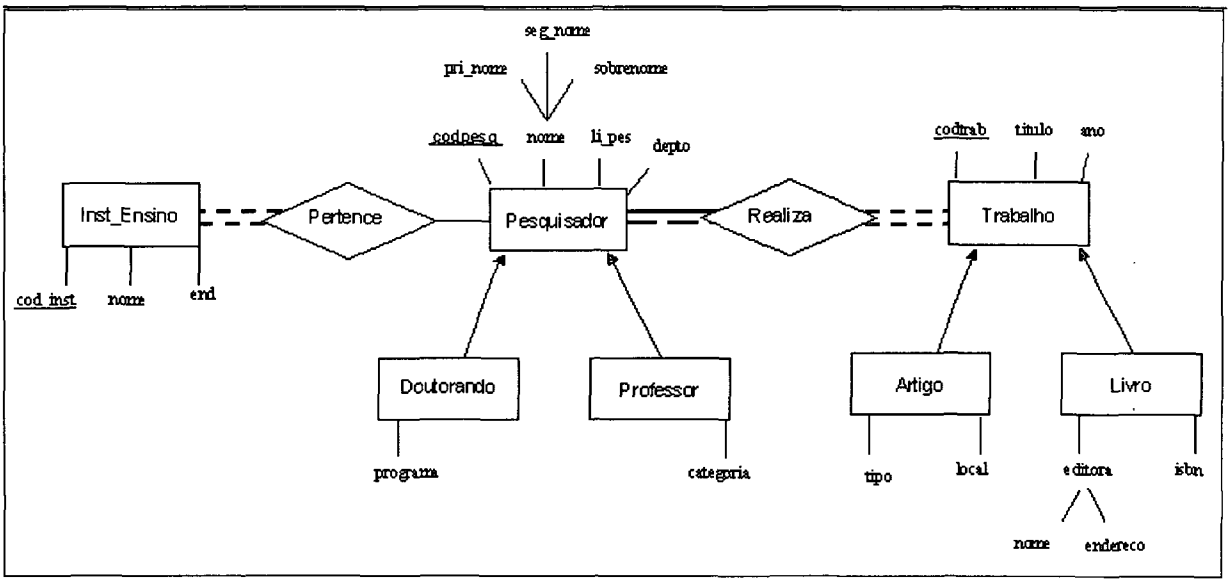


FIGURA 16 - DIAGRAMA ERC+ DO DOMÍNIO DA APLICAÇÃO
FONTE - Adaptado de (RODACKI, 2000)

Na figura 16 é apresentado o diagrama ERC+ do domínio da aplicação, onde são observados diversos conceitos do modelo ERC+, tais como a entidade Pesquisador ligada à entidade Trabalho pelo relacionamento Realiza. Este relacionamento pode ser expresso pelas seguintes interpretações 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, codtrab e cod_inst, aparecem sublinhados no esquema por serem atributos chaves das entidades Pesquisador, Trabalho e Inst_Superior, respectivamente.

6.4 ANÁLISE DAS TAREFAS DO USUÁRIO

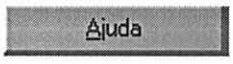
As tarefas apresentadas na tabela 2 identificadas como necessárias ao processo de especificação do predicado ou elaboração da consulta propriamente dita:

Elaboração da Consulta
Efetuar seleção de atributos para a resposta
Confirmar a escolha de atributos
Estabelecer Critérios Escolher atributo para comparação Inserir critério Selecionar critérios Aplicar operador lógico Confirmar a construção do predicado
Checar *feedback* da consulta em elaboração
Voltar à Etapa de Seleção do sub-esquema (Cancelar)
Solicitar Auxílio sobre a Etapa (Ajuda)
Finalizar a aplicação (Sair)

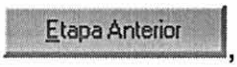
TABELA 2 -TAREFAS DO USUÁRIO NA ELABORAÇÃO DE CONSULTAS

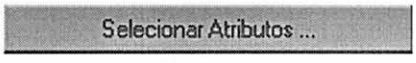
6.5 ESPECIFICAÇÃO E DESCRIÇÃO DA INTERFACE

Nesta fase da elaboração de predicados, o usuário já possui um sub-esquema ou esquema de trabalho determinado, que é um esquema visual produto da fase anterior onde o usuário seleciona, no esquema ERC+, quais os elementos que serão utilizados na elaboração da consulta (BASSI, 2002).

Como várias são as etapas a serem cumpridas para atingir o objetivo, é imprescindível que a interface possua mensagens, como aquelas veiculadas por um assistente, a guiar o usuário pela interface. Nas ferramentas avaliadas, sentiu-se muito a falta de expressividade, de mensagens que situassem o usuário no processo. Para suprir esta necessidade, em toda tela da interface proposta há mensagens orientando o usuário na execução das ações. Além disso, por meio do botão 

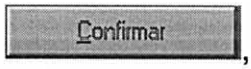
é ativado o auxílio do sistema (*help on-line*).

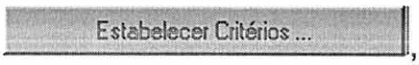
A partir do sub-esquema, sempre disponível para visualização, o usuário deverá escolher então, quais atributos farão parte da resposta, ou mesmo, retornar à etapa de seleção de sub-esquema, clicando no botão 

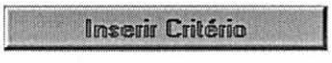
, cancelando a ação desta fase (*undo*)[3]. Caso o usuário opte por escolher atributos para a resposta, ele deverá clicar no botão 

para abrir um formulário específico para esta tarefa. A decisão de separar formalmente a parte de seleção da parte de estabelecimento de critérios, ativando esta última somente após a conclusão da primeira, está embasada na necessidade de estabelecer uma seqüência de ação no modelo mental do usuário, facilitando o processo de construção do predicado.

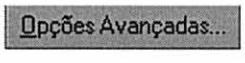
Para selecionar atributos para a resposta, a escolha é feita diretamente nas tabelas que formam o sub-esquema, pressionado-se o botão esquerdo do *mouse* com o cursor sobre seu nome e selecionando atributo(s). Se a seleção for simples, ou seja, de um único atributo, basta um clique sobre o atributo por meio do botão esquerdo do *mouse*. Para seleção de vários atributos da mesma tabela, a tecla

CTRL deve ser pressionada enquanto clica-se com o botão esquerdo do *mouse* com o cursor sobre os atributos desejados. Após seleção de todos os atributos necessários à exibição, o usuário deverá confirmar ação pressionando o botão 

, retornando à tela principal da etapa de Construção da Consulta, que, automaticamente, habilita o botão de comando 

, permitindo estabelecer valores ou intervalo de valores, fazer comparações entre atributos e construir expressões booleanas que servirão de parâmetros de comparação na execução da consulta. Esta sub-etapa exige que o usuário conheça a abstração do modelo ERC+, de forma que ele consiga expressar-se adequadamente para que o que ele deseja como resultado seja recuperado. Assim como na etapa anterior, o usuário clica por meio do botão esquerdo do *mouse* com o cursor sobre as tabelas disponíveis e, em seguida, clica com o cursor sobre o atributo desejado, selecionando-o. O ambiente orienta visualmente o usuário para que proceda executando os passos, da esquerda para a direita. Contudo, há uma certa liberdade na seqüência de passos para a criação da expressão relacional. Ele pode, por exemplo, atribuir um valor constante e depois definir o operador relacional de comparação. Ao clicar no botão 

o critério formulado será adicionado na Caixa de Construção dos Critérios. O usuário poderá incluir todos os critérios desejados e somente depois aplicar operadores lógicos para agrupá-los em uma única expressão, ou ir aplicando-os gradativamente, à medida que for inserindo os critérios, sempre de maneira pós-fixada e agindo sobre pares de expressões. Para aplicar um operador lógico, o usuário deverá selecionar o operador desejado e após, clicar no botão 

. Ainda nesta fase, o usuário poderá escolher em 

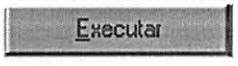
para aplicar os operadores redução e/ou simplificação à sua consulta. Estes operadores são descritos formal e detalhadamente na interface, incluindo exemplos de utilização disponibilizados no *help*.

---

[3] O rótulo Etapa Anterior foi escolhido em detrimento do rótulo padrão Cancelar devido a necessidade, identificada no processo de *design*, de o usuário ter ciência de que uma consulta é constituída de diversas etapas subsequentes.

As decisões descritas no parágrafo anterior estabelecem uma relativa flexibilidade da interface. Esta flexibilidade visa aumentar a usabilidade do sistema, bem como acomodar usuários novatos sem prejuízo dos usuários mais experientes.

O sistema exibe a janela de *feedback* das ações efetuadas, definindo as regras sintáticas de mapeamento e as restrições semânticas do sub-esquema. Este *feedback* é fornecido em uma sentença que expressa a consulta idealizada, na forma de pseudo-código em português – restrito à um padrão definido de forma a permitir ao usuário a verificação de se a consulta em elaboração é a realmente desejada, antes da efetivação da mesma. A sentença resultante expressará ao usuário o quê ele selecionou, o local de origem dos itens selecionados e as condições a serem respeitadas.

Neste ponto, o usuário poderá clicar no botão 

para efetivar sua consulta, passando à etapa de Visualização do Resultado.

É importante salientar que, a qualquer momento, o usuário poderá deixar o sistema, por meio do botão 

e que o sistema, além de mensagens de orientação em tela e ajuda *on-line*, disponibiliza várias janelas de alerta, com o intuito de viabilizar a correção de erros e orientar o usuário pela interface. Também, em todo o ambiente está ativado o recurso de ***hint*** (dica), conhecido do usuário no ambiente *Windows*, sensível ao contexto e ativado ao se posicionar o cursor do ***mouse*** sobre os componentes da interface.

6.6 DICIONÁRIO DE SIGNOS E REGRAS DE CORRELAÇÃO SEMÂNTICA (MAPEAMENTO SEMÂNTICO)

Para usuários com o estereótipo previamente descrito, o uso de ícones de metáforas para o mundo real dos elementos do esquema de banco de dados não é uma boa alternativa, pois os conceitos e termos técnicos envolvidos são amplamente difundidos e necessários para uso eficiente do ambiente. Desta forma, melhor é ensinar-lhe o jargão técnico ao invés de inserir uma dificuldade intermediária no

processo (a metáfora). Cabe aqui estabelecer relações de significante-significado entre os elementos presentes no ambiente de trabalho (signos de domínio) com qual o usuário interage e, na medida do possível, o seu significado para o usuário, objetivando construir um vocabulário específico da aplicação, mais especificamente, levantar os componentes verbais do *domain-register* (PRADO & BARANAUSKAS, 1999). *Helps* podem ser utilizados nesta tarefa para ajudar o usuário a adquirir tal conhecimento.

Do ponto de vista semiótico, expressões de signos podem ser identificadas e caracterizadas através de diversas substâncias, e cada uma dessas expressões tem associado um conteúdo. Assim, uma espécie de dicionário de signos pôde ser criado, como é mostrado na tabela 3:

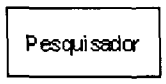
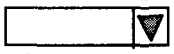
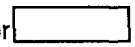
Objeto	Significado
	Objeto do Sub-esquema: Entidade
	Objeto do Sub-esquema: Relacionamento
	Objeto do Sub-esquema: Cardinalidade (1 ou mais)
	Objeto do Sub-esquema: Indicação de item não-obrigatório
	Objeto do Sub-esquema: Especialização/generalização
	Caixa de *Check-Box,* para escolha de Operação
	Caixa de *List-Box*, para a escolha de Operação
Valor 	Caixa de entrada para um valor constante

TABELA 3 - DICIONÁRIO DE SIGNOS

Para a construção do dicionário de signos inicial, foram observados os objetos do modelo ERC+, bem como os componentes básicos de interação disponíveis em ferramentas *Windows*. De posse da declaração dos signos do domínio da aplicação que estarão compondo o dicionário de signos e estarão presentes na interface para visualização e manipulação pelo usuário, é possível planejar um modelo de

interface, com todos os seus elementos, visando melhorar a interação usuário-sistema.

Um *widget* torna-se um signo quando sua aparência e comportamento na interface possuem um significado. Podem ser utilizados de diversas maneiras, no processo comunicativo.

Elemento da LEMD	*Widget* associado	Significado
View	Painéis ou *message-boxes*	Resultado apresentado pelo sistema referente ao acionamento de uma função da aplicação realizada pelo usuário
Text	*Labels* ou *message-boxes*	Mensagem direta do *designer* para o usuário
Activate	Botões de pressão	Acionamento de uma função da aplicação
Select information-of	*Combo-boxes* ou painéis	Escolha de uma opção de uma lista ou diagrama apresentado
Enter	Caixa de texto	Digitação da mensagem a ser enviada ao sistema
Join	Painéis ou *boxes*	Agrupamento
Repeat	Painéis ou *boxes*	A possibilidade da tarefa apresentada poder ser executada mais de uma vez
Sequence	Painéis ou *boxes*	Seqüência não obrigatória de execução das tarefas apresentadas
Combine	Painéis ou *boxes*	Combinação obrigatória e fixa de execução das tarefas apresentadas

TABELA 4 - CORRELAÇÃO SEMÂNTICA

6.7 EXEMPLOS DE CONSULTAS E DA UTILIZAÇÃO DO SISTEMA

Seguem consultas realizadas sobre o domínio de hipótese, procurando enfatizar aspectos relevantes neste processo, no que diz respeito à caracterização da interação usuário-sistema.

Exemplo 1: Deseja-se saber quem são os pesquisadores que trabalham com Banco de Dados e Interface Humano-Computador na UFPR.

Ao entrar no ambiente, figura 17, o usuário, tendo ciência dos dados que deseja obter como retorno, passa à fase de Seleção de Atributos, conforme figura 18.

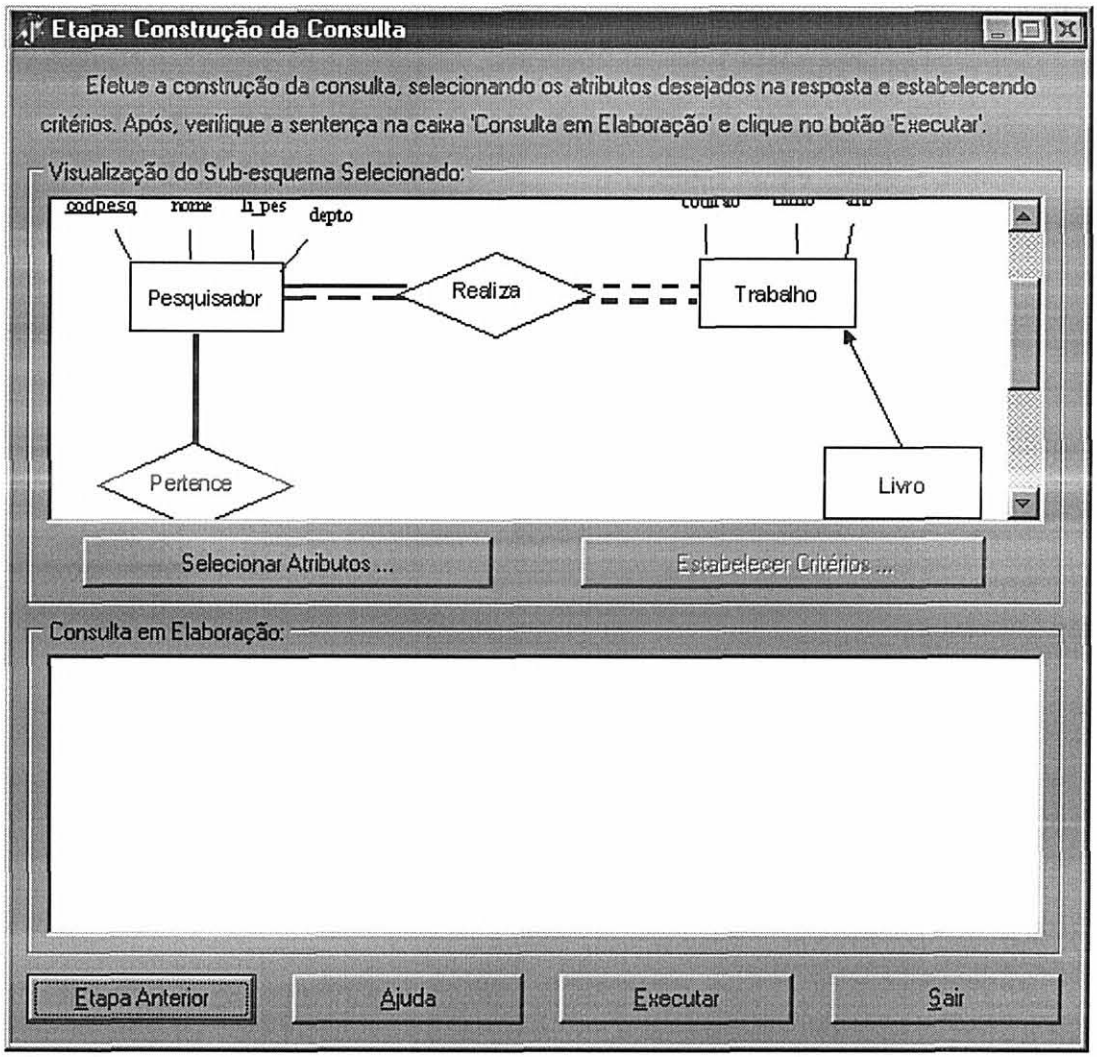


FIGURA 17 - TELA INICIAL DO AMBIENTE

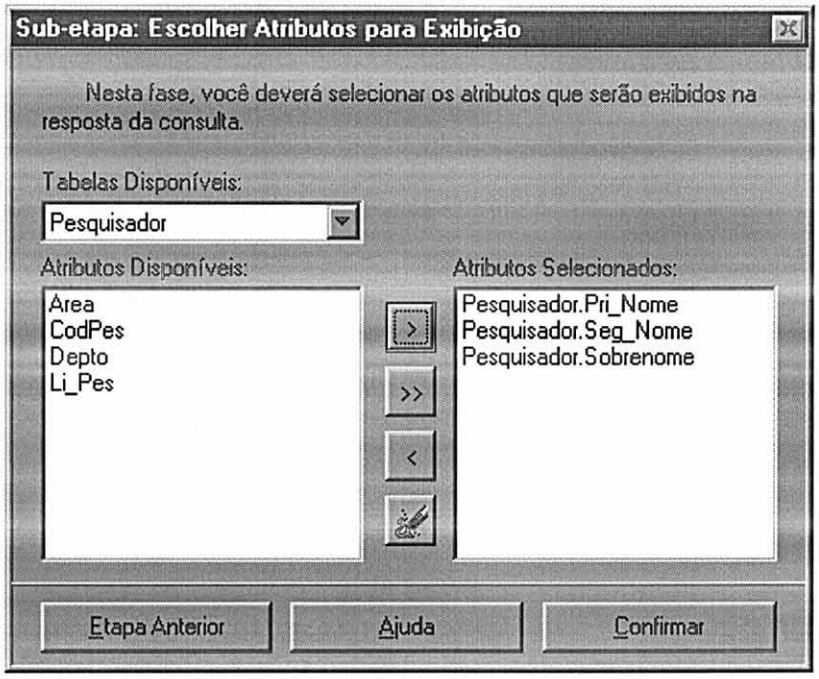


FIGURA 18 - ESCOLHA DE ATRIBUTOS

O usuário selecionará a tabela Pesquisador em Tabelas Disponíveis, selecionará os atributos Pri_nome, Seg_nome e Sobrenome, que compõem o nome completo. Esta seleção pode ser realizada individualmente por atributo e clique com o botão esquerdo no primeiro botão de seleção ou ainda, clicando-se sobre todos os atributos desejados na tabela, com o auxílio da tecla CTRL para então acionar o primeiro botão de seleção e após, clicar no botão 

. Deverá selecionar também o atributo nome da tabela Ins-Ensino.

A figura 19 trás os botões de seleção e suas respectivas funções:

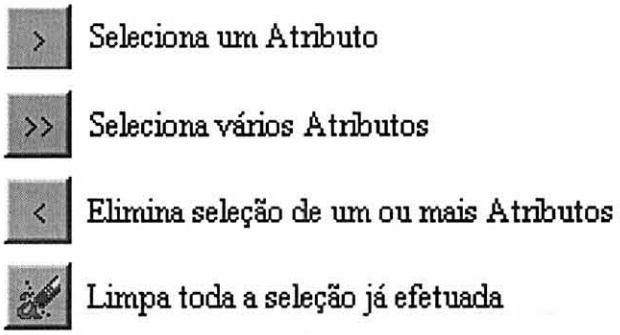


FIGURA 19 - BOTÕES DE SELEÇÃO E SUAS FUNÇÕES

Feito isto, retornará à tela inicial onde já estará na Caixa 'Consulta em Elaboração', uma sentença expressando as tarefas já realizadas e estará ativado o botão Estabelecer Critérios ... que inicialmente aparece desabilitado, expressando uma atividade não-disponível, como ilustrado na figura 20.

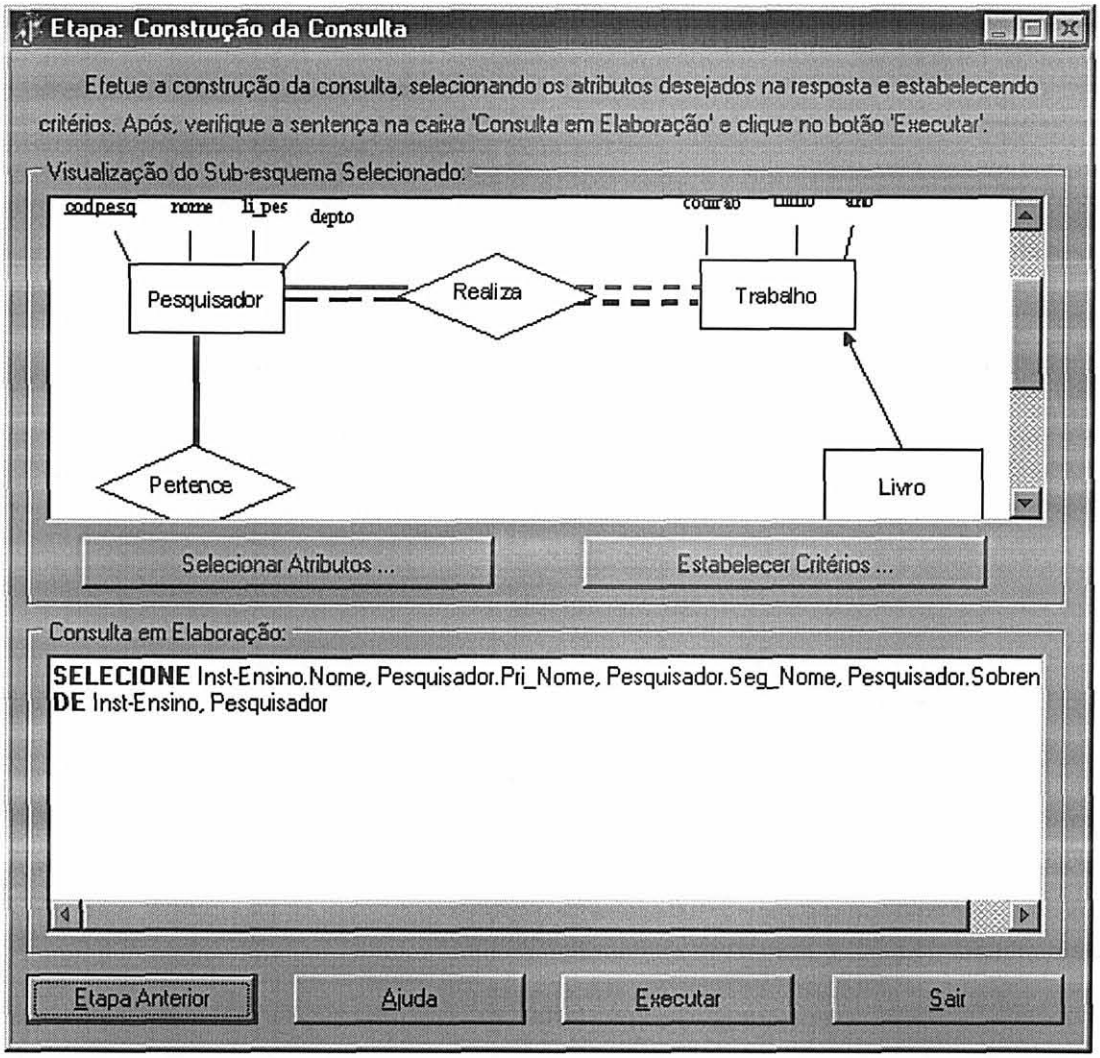


FIGURA 20 - TELA INICIAL COM SELEÇÃO DE ATRIBUTOS

Ao entrar na sub-etapa de estabelecimento de critérios, o usuário já possui algumas condições que deseja que sejam respeitadas: Selecionar professores de BD e IHC (tem que satisfazer ambas as condições. Um pesquisador que atue apenas em uma destas áreas não será listado na resposta). Ainda, há outra restrição, o mesmo tem que trabalhar na UFPR. Estas condições têm que ser expressas no sistema. Para saber que um professor é docente de uma Instituição de

Ensino Superior, há o relacionamento Pesquisador pertence à Instituição de Ensino e este caminho deve ser preservado. Porém, esta condição de junção está implícita ao usuário final, sendo efetuada pelo sistema, e não aparece no *feedback*. Os critérios de comparação devem ser inseridos diretamente nos atributos desejados, como expressa a figura 21. Esta comparação se dá por meio dos operadores relacionais próprios para este fim.

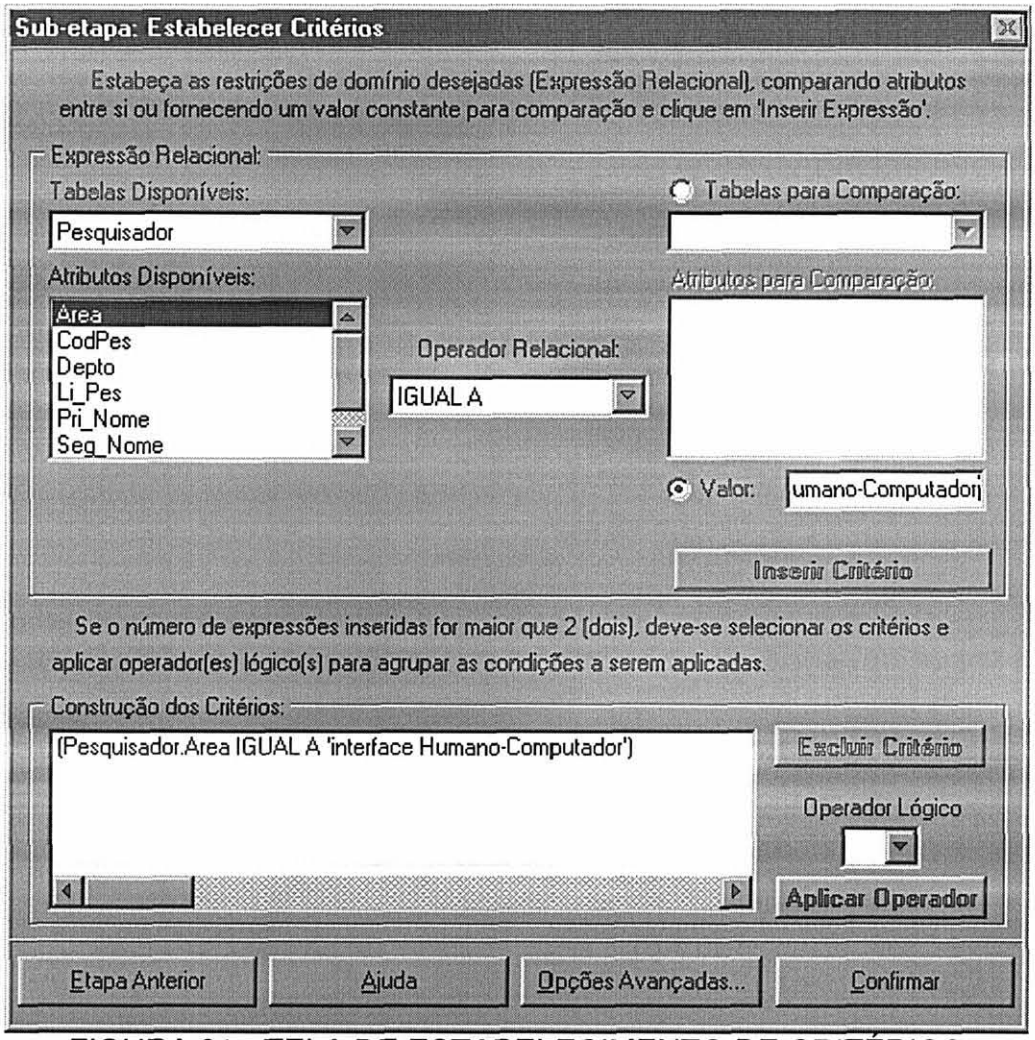


FIGURA 21 - TELA DE ESTABELECIMENTO DE CRITÉRIOS

Os demais critérios deverão ser inseridos de forma análoga. Após todos os critérios estabelecidos, entra-se no processo de aplicar operador(es) lógico(s). Esta necessidade dá-se pelo fato de que todas as condições são passadas ao processamento da consulta como um único predicado agrupado (caso haja mais que um critério de restrição). Selecionam-se os critérios e, aplica-se operador lógico

desejado. Estes, são transformados em um único, que deverá ser novamente agrupado, até que exista uma única expressão de consulta.

O usuário tem a liberdade no sistema de primeiramente inserir todos os critérios e somente após, agrupando-os dois a dois, aplicar o operador lógico desejado, como demonstra a figura 22, ou ir aplicando o operador desejado a medida que possuir pares de critérios inseridos.

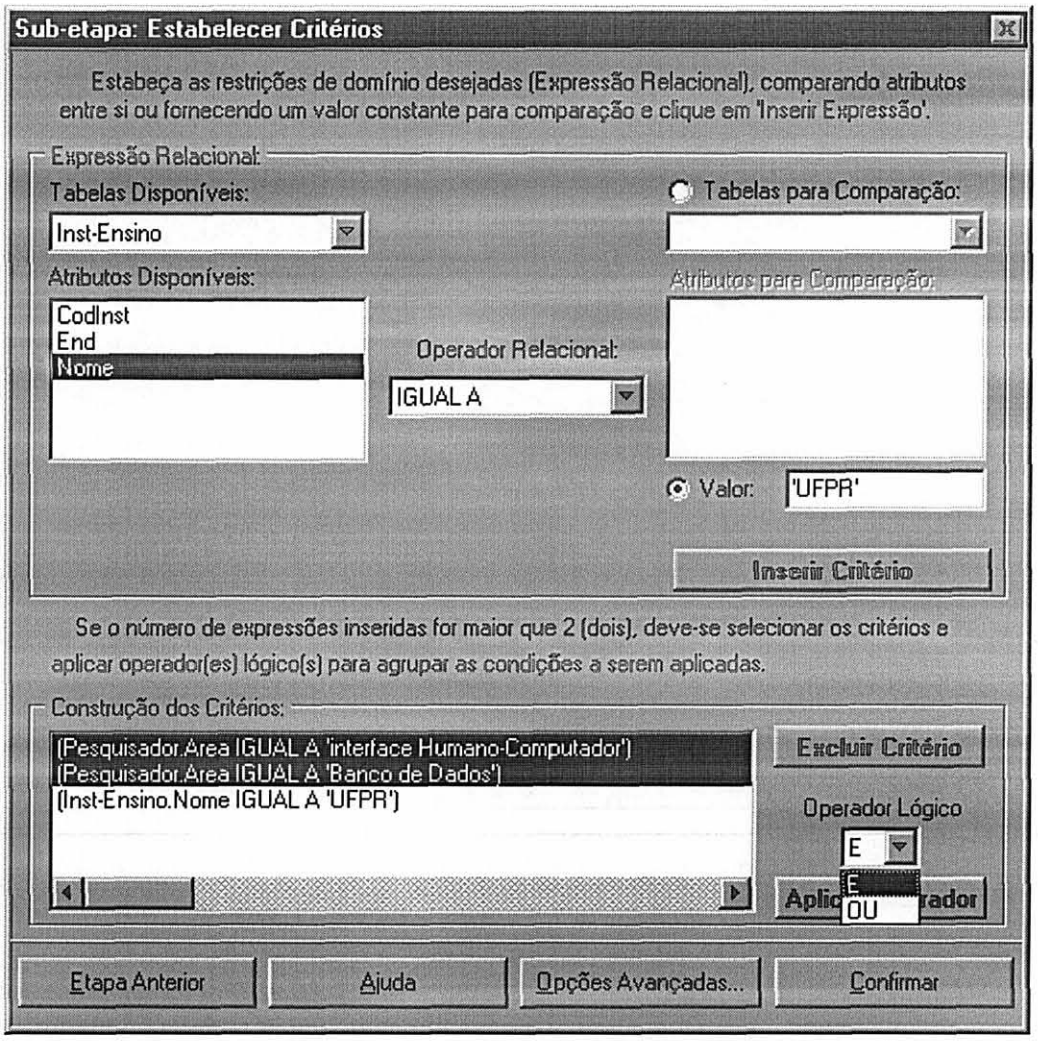


FIGURA 22 - TELA COM APLICAÇÃO DE OPERADOR LÓGICO

Após a confirmação, retorna-se à tela principal, onde, o critério construído já está agrupado ao resultado da etapa de seleção dos atributos, formando uma expressão de consulta expressa ao usuário em alto nível. A abordagem utilizada

estabelece uma hierarquia de parênteses aninhados, expressando inclusive, a ordem de execução das comparações. Optou-se por esta metodologia devido ao fato de que uso de parênteses é totalmente familiar a qualquer usuário de aplicação. A tela da figura 23 exibe a sentença completa da consulta em elaboração, pronta para a execução. Clicando-se no botão Executar, passa-se então à etapa de visualização da resposta.

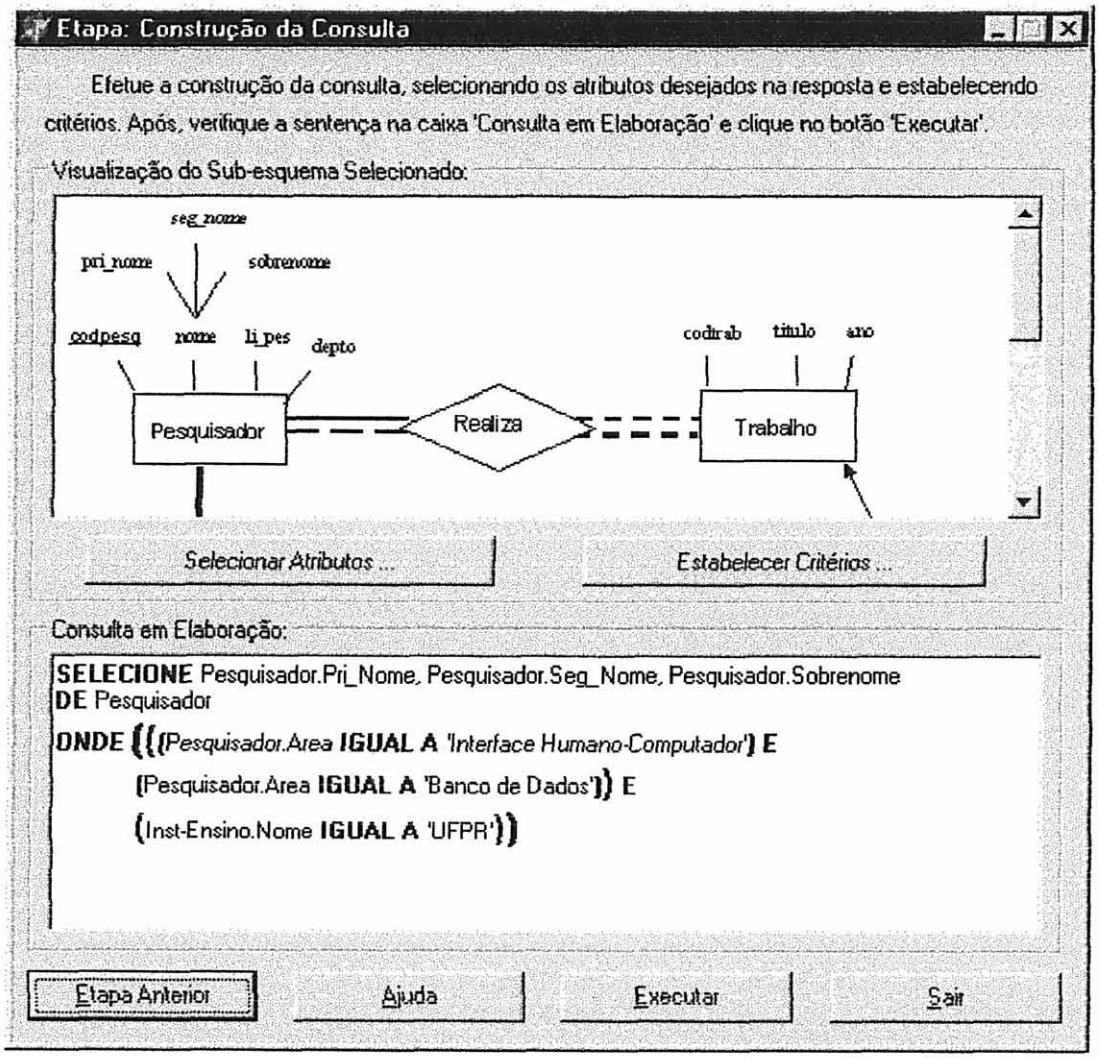


FIGURA 23 - TELA PRINCIPAL COM CONSULTA ELABORADA

Exemplo 2: Deseja-se selecionar o nome completo de todos os pesquisadores e o título de seus trabalhos publicados durante e após o ano de 2000. Porém, caso o

pesquisador não possua publicações no período especificado, ele também deverá constar na resposta da consulta.

O caso agora exemplificado parte de uma consulta elaborada seguindo as fases recém apresentadas no exemplo 1, com o intuito de apresentar o funcionamento da interface na aplicação do operador de redução.

Deve-se notar que para a elaboração desta consulta o mesmo sub-esquema de trabalho exibido na figura 17 é utilizado. Esta é uma vantagem que o ambiente fornece, de aproveitar todo o potencial de um sub-esquema selecionado, reaproveitando-o, de forma a evitar que o usuário tenha que manipular o esquema global, original, para refazer a etapa de seleção de sub-esquemas. As telas das figuras 24, 25 e 26 exprimem os processos de seleção de atributos e estabelecimento de critérios para a consulta expressa no exemplo 2.

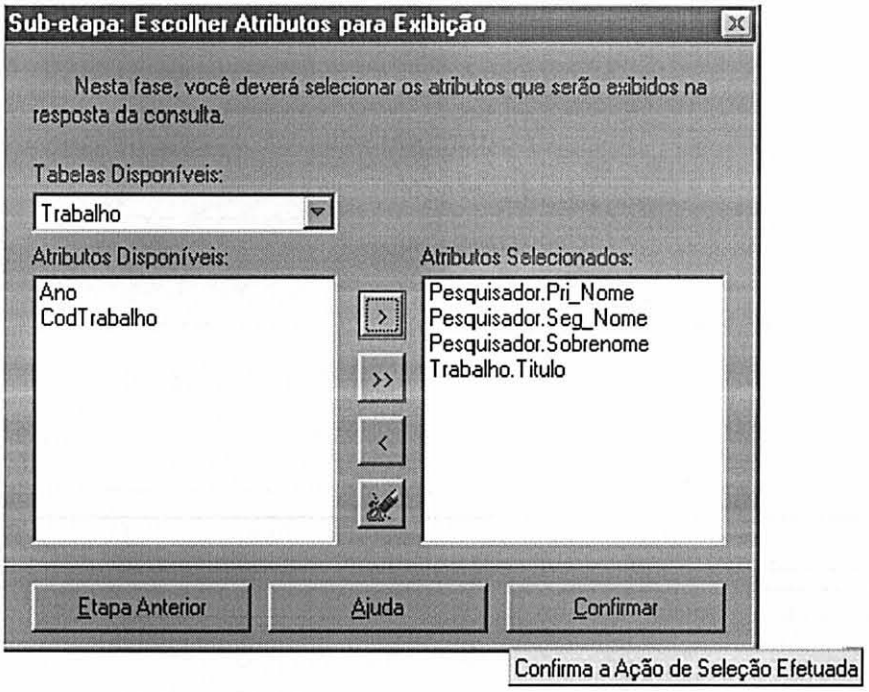


FIGURA 24 - TELA DE SELEÇÃO DE ATRIBUTOS PARA O EXEMPLO 2

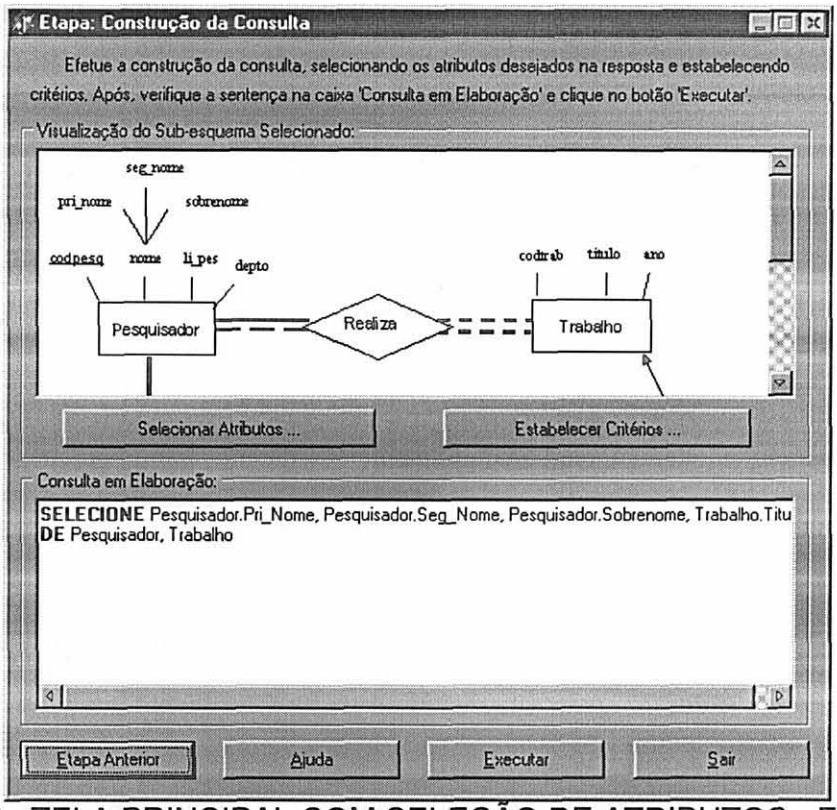


FIGURA 25 - TELA PRINCIPAL COM SELEÇÃO DE ATRIBUTOS - EXEMPLO 2

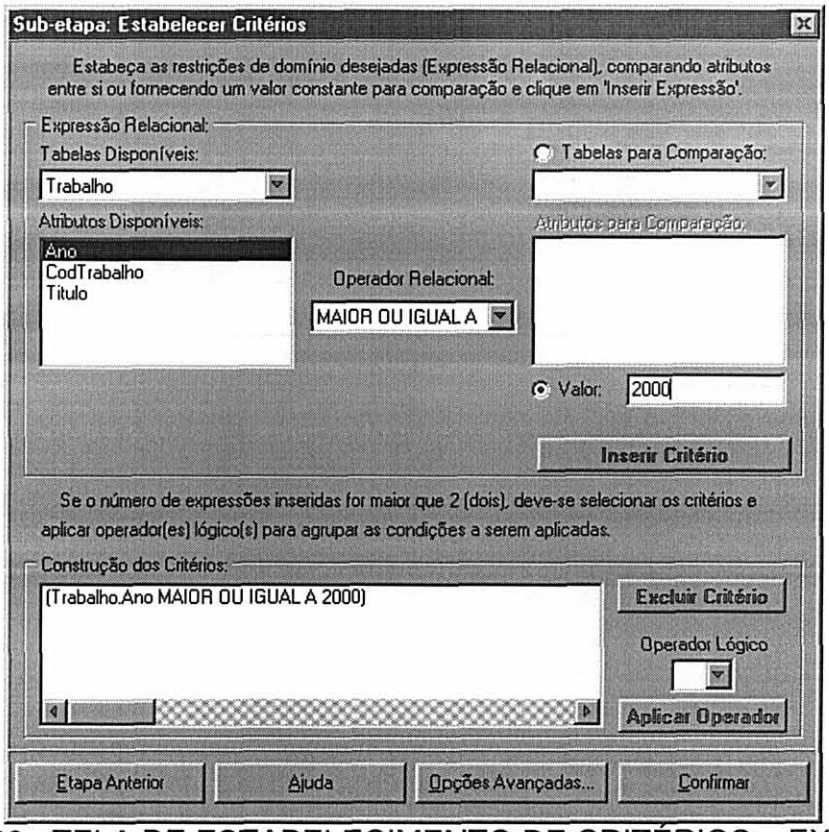
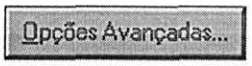


FIGURA 26 - TELA DE ESTABELECIMENTO DE CRITÉRIOS – EXEMPLO 2

Com a realização das tarefas recém mostradas, o resultado da consulta não procederia ao desejado pelo usuário pois apenas seriam selecionados os pesquisadores que atendessem à condição de comparação estabelecida no atributo Ano da tabela Trabalho. Ou seja, as pessoas que não obtiveram publicações a partir do ano de 2000 não iriam constar na resposta.

Para que todos os pesquisadores possam aparecer no resultado da consulta de forma que a não-satisfação da condição imposta não implique a exclusão do conjunto resposta, o operador de redução deve ser acrescentado ao predicado. Para isto, na sub-etapa de estabelecimento de critérios, o usuário deverá clicar no botão esquerdo do *mouse* com o cursor sobre o botão de comando 

e então marcar a opção Aplicar Operador de Redução, como descreve a figura 27.

O uso eficiente deste operador, bem como do operador de simplificação, requer que o usuário os compreenda e identifique situações em que eles possam ser aplicados. Por tratar-se de conceitos abstratos, que o usuário compreenderá ao utilizar a ferramenta, o auxílio *on-line* traz exemplos e explicações detalhadas.

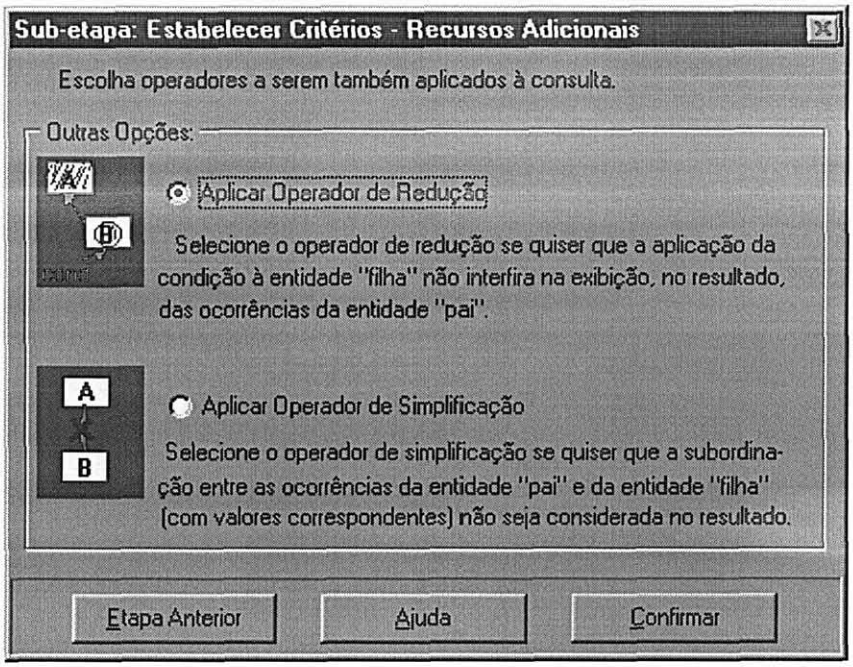


FIGURA 27 - TELA DE RECURSOS AVANÇADOS – EXEMPLO 2

A figura 28 exibe a tela principal do ambiente, com a consulta que satisfaz ao exemplo 2, pronta para a execução.

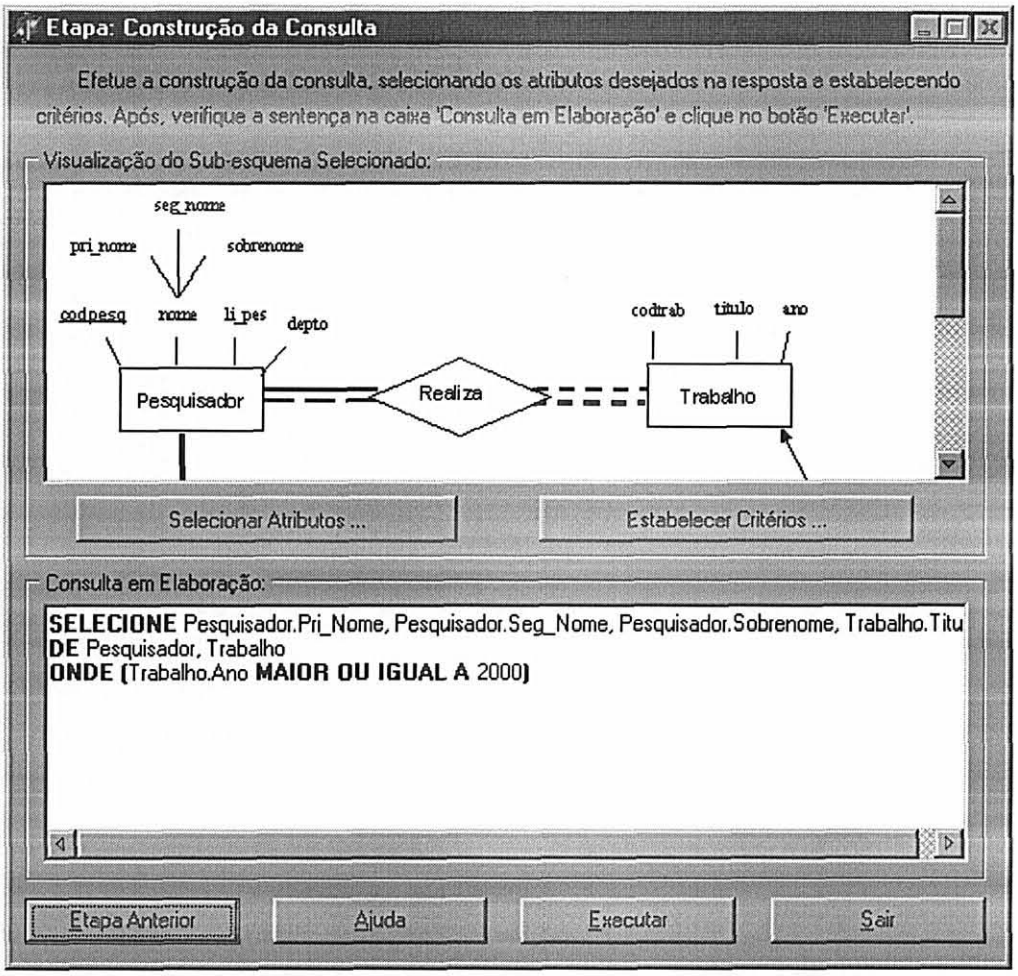


FIGURA 28 - TELA PRINCIPAL COM A CONSULTA DO EXEMPLO 2

7 ANÁLISE DO AMBIENTE DE INTERFACE PROPOSTO

Para o *design* deste ambiente, partiu-se dos princípios básicos de usabilidade que envolvem, segundo (ROMANI & BARANAUSKAS, 1998), três categorias principais: facilidade com que usuários novos podem efetivamente começar a interagir e alcançar máximo desempenho; multiplicidade de maneiras com que o usuário e o sistema trocam informação e nível de suporte que o usuário tem para determinar seu sucesso e fazer a avaliação de suas metas.

No que diz respeito à primeira categoria, procurou-se revelar ao máximo os componentes integrantes do esquema ERC+, necessários à utilização do ambiente, bem como as funções que o sistema disponibiliza. Para enquadrar-se na segunda categoria, a flexibilização no processo de Estabelecimento de critérios, embora sendo um processo com início, meio e fim, não se limitou à interação por uma forma rigidamente preestabelecida mas, ofereceu-se ao usuário algumas maneiras de agir, de forma que este possa optar pela que melhor lhe atenda. Ainda, houve a preocupação de explicar na interface os operadores de redução e simplificação, de forma auxiliar o usuário na sua utilização. Já no tocante à terceira categoria, as mensagens de orientação fornecidas pela interface, as janelas de alerta, o *hint* ativado em todos os signos da interface e um auxílio *on-line* foram construídos para atendê-la.

A exigência ao usuário de conhecimento em banco de dados, senão prévio fornecido pelo sistema, é hipótese do trabalho para garantir a ciência do mesmo na construção das consultas a partir de ações de manipulação direta sobre o esquema. Esta necessidade determina a adoção de um sistema completo de auxílio.

Embora nenhuma avaliação heurística tenha sido aplicada à ferramenta, tentou-se respeitar o conjunto de heurísticas desenvolvidas por (NIELSEN, 1993 *apud* ROMANI & BARANAUSKAS, 1998), sobre problemas de usabilidade, de forma a tentar evitá-los ou, no mínimo, minimizá-los. Estas heurísticas de usabilidade foram respeitadas com base na apropriação dos conceitos e no uso de técnicas da Engenharia Semiótica.

Segue o supracitado conjunto de heurísticas descrito em (ROMANI & BARANAUSKAS, 1998), com colocações sobre a maneira em que o ambiente as considera:

- **Estética e *design* mínimos**: Os diálogos não devem conter informações irrelevantes ou que sejam raramente necessárias, o que comprometeria a visibilidade.
- **Coerência entre o sistema e o mundo real**: O sistema deve falar a língua do usuário, com palavras, frases e conceitos familiares ao usuário, ao invés de termos orientados ao sistema. As informações devem ter uma ordem lógica e natural.

Procurou-se minimizar as expressões do domínio de banco de dados, subindo a um nível mais focado na aplicação do domínio. Contudo, a forte dependência do domínio de banco de dados é um fator limitante, que deve ser contornado pelo projeto de comunicação.

- **Reconhecer é melhor que lembrar**: O sistema deve fazer objetos, ações e opções visíveis. Instruções para usar o sistema devem ser visíveis ou fáceis de serem recuperadas sempre que necessárias.
- **Consistência e padronização**: usuários não devem ter surpresas se diferentes palavras, situações ou ações significam a mesma coisa. Deve-se seguir uma plataforma de convenções.
- **Visibilidade do *status* do sistema**: O sistema deve sempre manter os usuários informados sobre o que está sendo feito, através de um *feedback* apropriado, em tempo razoável.

O ambiente fornece a janela de *feedback* 'Consulta em Elaboração', além de mensagens de alerta para situações inusitadas.

- **Controle e liberdade de ações**: Usuários freqüentemente escolhem funções erradas e necessitam de clareza nas opções de "saídas de emergência" sem ter que atravessar um extenso diálogo. O sistema deve prover *undo* e *redo*.

O usuário tem opções no sistema para retomar atividades (quando retorna à opções de seleção de atributos ou de definição de critérios) ou mesmo cancelá-las

(por meio de botões específicos como o Limpar em Seleção de Atributos, Excluir Critérios na sub-etapa Estabelecimento de Critérios ou até mesmo, a opção Etapa Anterior, que possui a função de cancelar a etapa corrente).

- **Flexibilidade e eficiência de uso**: O sistema deve acomodar usuários novatos e experientes. O sistema deve permitir ao usuário ajustar suas ações freqüentes.
- **Ajuda aos usuários a reconhecer, diagnosticar e recuperar erros**: Mensagens de erros devem ser expressas em linguagem plana, sem códigos, indicando o problema e sugerindo soluções.
- **Previne erros**: A presença de mensagens explicativas deve ser um cuidado de *design* que previne a ocorrência de problemas.
- **Documentação e Ajuda**: Qualquer informação deve ser facilmente encontrada e enfocar a tarefa do usuário. Uma lista de passos concretos deve ser disponibilizada e não deve ser extensa.

Foram também consideradas a LEMD, que permitiu a construção do programa descrito no apêndice A, para o desenvolvimento da ferramenta para a determinação do controle das mensagens, que traduzem o modelo de interface como expressão da mensagem do *designer*, e a STAG, detalhada no apêndice B, na especificação de primitivas de interação da linguagem visual.

Pesquisando-se metáforas para o mundo real em um domínio de banco de dados visuais, concluiu-se que os elementos do modelo são, por definição, abstratos e devem ser compreendidos para uso adequado. Então, muitos dos elementos que compõem os sub-ambientes fazem uso de termos técnicos ao invés de ícones que representem uma metáfora do mundo real. Esta decisão considera que o uso de ícones acrescentaria um novo signo para interpretação do usuário (o ícone), não muito expressivo, e que remeteria o usuário novamente aos termos técnicos da área em questão. Quanto às fases na elaboração de consulta, procurou-se estabelecer na interface um processo lógico de execução.

Como no SUPER, a interface de consulta é consistente com a definição de dados da interface, ou seja, a interação com os usuários é baseada em um conjunto comum de conceitos de modelagem de dados.

A seguir, são discutidas as principais características deste ambiente no tocante a cada uma das sub-fases da elaboração da consulta identificadas.

7.1 SUB-FASE: SELEÇÃO DE ATRIBUTOS

A seleção de atributos é uma etapa completamente estruturada, pois o usuário escolhe as entidades envolvidas e os respectivos atributos por meio do clique do *mouse* e botões de seleção. Ainda, pode-se anular uma ação realizada, por meio de botão próprio.

Embora esta fase também exija conhecimento sobre o modelo de dados especificado, devido à natureza simplificada da ação na interface, dificilmente acarretará erros mas, se isto acontecer, estes poderão ser facilmente corrigidos.

7.2 SUB-FASE: ESTABELECIMENTO DE CRITÉRIOS

Esta fase é estritamente dependente dos conceitos do domínio de banco de dados, devendo o usuário, para seu uso eficiente, estar familiarizado com a abstração do modelo (devido à necessidade de trabalhar também sobre os relacionamentos e não apenas sobre as tabelas apresentadas no diagrama), bem como conhecer as restrições semânticas impostas pelo domínio da aplicação.

O ambiente contribui para a minimização destes problemas, como evidenciado abaixo.

7.2.1 DEFINIÇÃO DE EXPRESSÕES RELACIONAIS

Nesta etapa, procurou-se dispor a caixa de opção com os operadores relacionais disponíveis em uma posição na interface que favoreça interpretar a aplicação dos mesmos. Procurou-se também, deixar aberta a possibilidade de o

usuário inserir as restrições que julgar necessárias à consulta, estabelecendo comparação, podendo fazê-la com outros atributos disponíveis na base de dados (respeitando relacionamentos do Diagrama ERC+) ou fornecendo valores por meio da Caixa de Entrada 'Valor'.

7.2.2 CONSTRUÇÃO DE PREDICADOS COM OPERADORES LÓGICOS

Este é um ponto crítico em todas as interfaces visuais para banco de dados revisadas, devido à forte relação com o raciocínio lógico obtido pelo usuário durante toda a concepção de sua formação intelectual.

SHNEIDERMAN (1993) afirma que muitas pesquisas e experiências com sistemas de recuperação booleana indicam clara e repetidamente que a sintaxe de formulação e técnicas de recuperação booleana não são muito eficientes na busca e nem muito usáveis ou eficientes em métodos de consulta para usuários finais. SHNEIDERMAN afirma ainda que, para usuários formularem de forma efetiva uma consulta nesta abordagem, eles devem estar muito familiarizados com a lógica booleana.

Em (PRATT & MURAMATSU, 2001) é relatado que a transformação de operadores booleanos em consultas resulta no não-entendimento pelo usuário final, por este não compreender os operadores e não conseguir formular o modelo de resposta do sistema.

Uma abordagem baseada em fluxo de dados, onde uma representação contínua eqüivale a uma operação *AND* e uma representação não contínua expressa uma operação *OR* é apresentada em (SHNEIDERMAN, 1993). Também (MURRAY *et al.*, 1998) em suas pesquisas desenvolveu um modelo visual, 3-D, igualmente baseado nesta abordagem de fluxo de dados, fazendo analogia com encanamentos (tubulações).

A idéia de estabelecer a ordem de comparação dos critérios com base em parênteses está intimamente ligada a uma estrutura de hierarquia que os parênteses sempre representaram em nossa cultura.

Contudo, nesta fase, em todas as abordagens estudadas e, também na proposta, o usuário precisará de um treinamento, além da utilização de documentação, para obter exemplos e formular sua correlação de expressão-significado.

7.3 O *FEEDBACK* FORNECIDO

O retorno como uma sentença estruturada, em alto nível, parece uma solução interessante, pois, além de expressar em uma linguagem próxima ao português, demonstra o raciocínio estruturado. A idéia é expressar "O que se deseja?" por meio de SELECIONE, "De onde advém a informação requisitada?" por meio da palavra DE e "Quais restrições deve-se impor?" por meio da palavra ONDE (todos os elementos após esta cláusula são critérios estabelecidos, incluindo operadores relacionais e/ou lógicos.

7.4 CONSIDERAÇÕES TÉCNICAS

O ambiente foi prototipado em *Borland Delphi* 6.0 e, para o desenvolvimento do ambiente de ajuda, foi utilizada a ferramenta *HelpScribble*.

8 CONCLUSÕES

8.1 CONTRIBUIÇÕES

Este trabalho realizou o desenvolvimento formal e sistêmico de um ambiente visual de elaboração de consultas.

Uma das contribuições principais deste trabalho está na especificação da interface em nível abstrato, permitindo a representação de associações entre intenções de comunicação do *designer* de interfaces visuais de consulta a banco de dados e expressões na interface utilizada para este fim. A aplicação da LEMD foi elaborada sobre todo um ambiente de interface, dando um passo à frente em sua utilização, visto que em (PRADO & BARANAUSKAS, 2000), o método de avaliação tinha sido proposto em um conjunto de janelas que foram isoladas e trabalhadas com certa independência. Este formalismo foi utilizado como instrumento de auxílio ao atendimento dos requisitos determinados pela abordagem cognitiva.

O poder de expressão da interface em relação a revelação das escolhas feitas pelo projetista foi aumentado em relação ao VIQUEN, pois, além de preocupar-se com a comunicação usuário-interface, incluindo a exibição da consulta em elaboração de forma mais natural ao usuário, a exibição de mensagens de orientação e de alerta ao longo de todo o processo, um sistema completo e interativo de ajuda *on-line*, mensagens de alerta, a não-disponibilização de expressões SQL e a explicação dos operadores redução e simplificação da álgebra ERC+, que em VIQUEN eram apenas apresentados na interface para seleção, sem qualquer explicação sobre a sua funcionalidade.

Na descrição do dicionário de signos da interface, ficou nítida a necessidade de que o usuário tivesse um domínio mínimo da área de Banco de Dados, para que conseguisse interpretar os signos apresentados na interface. O uso de assistente para esta tarefa é mais um facilitador da interação.

O processo de interação tornou-se mais consistente, pois vai habilitando as opções à medida que o usuário avança em sua tarefa de construção de consultas.

A exigência do conhecimento do domínio de banco de dados pelo usuário e a correspondente necessidade de orientação explícita e sob demanda se constituiram em uma decisão de projeto que priorizou no estado da arte dos ambientes de consultas visuais (de grande complexidade), a comunicabilidade da interface. Neste enfoque, o trabalho contribui com uma parte da solução do problema, revelando, contudo, a necessidade de continuidade no tratamento do problema de forma a alcançar soluções mais satisfatórias capazes de prescindir ou minimizar o papel crucial que o sistema de ajuda tem na solução ora apresentada.

Uma sessão de treinamento aos usuários do sistema, principalmente sobre a aplicação de operadores booleanos, para que o usuário obtenha exemplos e possa construir mais rapidamente a sua correlação expressão-significante é oferecida adicionalmente pela *interface proposta*.

8.2 TRABALHOS FUTUROS

Como este trabalho não tem a pretensão de ter esgotado o tema, esta pesquisa continua, no sentido de desenvolver uma interface que efetivamente expresse toda a meta-comunicação e permita a usabilidade necessárias ao ambiente de hipótese. Nesta direção, são apresentadas as seguintes sugestões:

- Integração da ferramenta VIQUEN com esta e com as todas as propostas de interface que compõem o presente projeto (BASSI, 2002), RANTHUM, 2002), bem como utilizando uma ferramenta própria para construção de objetos para manipulação direta, sugerindo-se a ferramenta ILOG, utilizada no VIQUEN.
- Há a necessidade de um aprofundamento no estudo de metodologias de avaliação nas vertentes Cognitiva e Semiótica para propor novos métodos que complementem a LEMD, usada no processo construtivo.

- Aplicação de métodos de avaliação à ferramenta completa, diferentes da LEMD, de forma a identificar possíveis problemas que ainda dificultem a interação.
- Avaliação da ferramenta por meio de testes de utilização por diferentes tipos de usuários, visando obter uma avaliação estatística (percentual de acertos, por exemplo) sobre o modelo de usabilidade.

9 REFERÊNCIAS BIBLIOGRÁFICAS

ADAMS, A. **Discoverer 4I: A Best Practices Case Study**. Disponível em http://www.avanco.com/discoverer_4i.htm Acesso em: 07 fev, 2002.

ANGELACCIO, M.; CATARCI, T.; SANTUCCI, G. **QBD*: A Graphical Query Language With Recursion**. IEEE Transactions on Software Engineering, Vol. 16, N° 10, 1990, 1150-1163.

ANGELACCIO, M.; CATARCI, T.; SANTUCCI, G. **QBD*: A Fully Visual Query System**. Journal of Visual Languages and Computing, Vol. 1, No. 2, p. 255-273, 1990.

BASSI, P. R. de. **Um Ambiente de Interface Visual para a Geração de Sub-Esquemas para uma Ferramenta de Consulta baseada no Modelo ERC+**. Dissertação de Mestrado. Departamento de Informática. Setor de Ciências Exatas, Universidade Federal do Paraná. Curitiba, 2002.

BATINI, C.; CATARCI, T.; COSTABILE, M.; LEVIALDI S. **Visual Query Systems – Technical Report** – Dipartimento di informatica e Sistemistica, Universitá di Roma "La Sapienza", 1992.

BECKER, K.; CARDOSO M. O. **Mail-by-Example: A Visual Query Interface for Managing Large Volumes of Data**. Anais do XV Simpósio Brasileiro de Banco de Dados. João Pessoa, 2000.

BRENNAN, S.E. **Conversation as Direct Manipulation: An Iconoclastic View**. In: B.K. Laurel, ed. **The Human-Computer Interface Design**. Reading MA: Addison-Wesley, 1990.

BOSCARIOLI, C.; GARCÍA, L. S.; RANTHUM, G.; BASSI, P.; SUNYE, M. S. **Visual Databases Languages: A tool evaluation based on the Designer's Message Specification Language**. Annales La conférence NîmesTIC. Nîmes, França, 2001.

CATARCI, T.; COSTABILE M. F.; LEVIALDI S.; SANTUCCI G. **A Graph-Based Framework for Multiparadigmatic Visual Access to Databases**. IEEE, 455-475, junho,1996.

CATARCI, T.; COSTABILE M. F.; MASSARI, A.; SALADINI, L.; SANTUCCI, G. **A Multiparadigmatic Visual Environment for Adaptive Acess to Databases**. ACM SIGCHI Bulletin, special issue on The Role of HCI in Italy, Vol. 28, N. 3, pp. 89-96, July 1996.

CHEN, P. **The Entity-Relationship Model – toward a unified view of data.** ACM: Transactions on Database Systems, v. 1, n. 1, p. 9-36,1976.

COHEN, P.R. **The Role of Natural Language in a Multimodal Interface**. Technote 514, SRI International, menlo Park, 1991.

DENNENBOUY, Y. **Flexibility of Visual Languages for Data Manipulation**. IFIP WG2.6 3rd Working Conference on Visual Database Systems, Março-1995.

ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. The Benjamim/Cummings Publishing Company, Inc., 2000.

FERNANDES, D.; SALGADO A. C. **GeoVisual Interface: A Visual Query Interface for Geographic Information Systems**. Anais do XV Simpósio Brasileiro de Banco de Dados. João Pessoa, 2000.

GARCÍA, L. S. **Interação Humano-Computador**. Departamento de Informática, Universidade Federal do Paraná. Curitiba, 2000.

GUEIBER, E. **VIQUEN – Um Ambiente Interativo para Consulta Visual e Extração de Esquemas**. Dissertação de Mestrado. Departamento de Informática. Setor de Ciências Exatas, Universidade Federal do Paraná. Curitiba, 2001.

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

LEITE, J. C. **Modelos e Formalismos para a Engenharia Semiótica de Interface de Usuário.** Tese de Doutorado. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 1998.

MARTINS, I.H. **Um Instrumento de Análise Semiótica para Linguagens Visuais de Interfaces**. Tese de Doutorado. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 1998.

MORAN, T. **The Command Language Grammars: a represetantion for the user interface of interactive computer systems**. International Journal Of Man Machine Studies, 15, 3-50, 1981.

MURRAY, N.; NORMAN P.; GOBLE C. **A Visual Query Language for Object Databases**. Manchester, 1998. Disponível em: <http://www.cs.man.ac.uk/~murrayn> Acesso em 06 jan. 2002.

_____. **A 3D Environment for Querying ODMG Compliant Databases**. Manchester, 1998. Disponível em: <http://www.cs.man.ac.uk/~murrayn> Acesso em: 15 abr. 2000.

NORMAN, Donald A. in collaboration with BROWN, Gill M. **Cognitive Models in Human-Computer Interaction**, In Norman, D. A., Draper, S. W., **User Centered System Design**. Hillsdale/NJ. Lawrence Erlbaum Associates, Cap. 3, 1986.

ÖZSOYOGLU G.; WANG H. **Example-Based Graphical Database Query Languages**. IEEE Computer Society, volume 26, número 5, 25-38, maio 1993.

PAYNE, S. J. ; GREEN, T. R. G. **Task-Action Grammars: A Model of the Mental Representation of Task Languages**. In Human-computer Interaction, Volume 2, pp 93-133, 1986.

PORTO, P. R. P. **Bancos de Dados com Interfaces Inteligentes**. Dissertação de Mestrado. Instituto de Informática. Universidade Federal do Rio Grande do Sul. Porto Alegre, 1997.

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.

PRADO, A. B.; BARANAUSKAS, M. C. C. **Projeto Granel – Investigando Possibilidades da Abordagem Semiótica em Design de Interfaces**. Atas do II Workshop sobre Fatores Humanos em Sistemas Computacionais. IHC'1999, Campinas/SP, 1999.

PRADO, J. P. A. **Tópicos de Interface Homem-Máquina**. UNESP - São Paulo: 1993.

PRATES, R.O.; LEITE J.C.; SOUZA C.S. de **Semiotically-Based User Interface Design Languages**. IHC'98, 18-27, 1998.

PRATT, W.; MURAMATSU, J. **Transparent Queries: Investigating User's Mental Models of Search Engines**. Information & Computer Science. University of California, Irvine, 2001.

RANTHUM, G. **Um Ambiente Visual de Interface para a Apresentação da Resposta para uma Ferramenta de Consulta baseada no Modelo ERC+.** Dissertação de Mestrado. Departamento de Informática. Setor de Ciências Exatas, Universidade Federal do Paraná, Curitiba, 2002.

ROCHA, H. V. da; BARANAUSKAS, M. C.C. **Design e avaliação de interfaces humano-computador**. São Paulo, 2000.

RODACKI, A. **Aplicação de Estratégias de Integração de Banco de Dados: um Estudo de Caso**. Dissertação de Mestrado. Departamento de Informática. Setor de Ciências Exatas, Universidade Federal do Paraná, Curitiba 2000.

ROMANI, L. A S.; BARANAUSKAS, M. C. C. **Avaliação Heurística de um Sistema altamente dependente do domínio**. Relatório Técnico. Instituto de Computação. UNICAMP, 1998.

SHNEIDERMAN, B.; YOUNG, D. **A Graphical Filter/Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation**. Human-Computer Interaction Laboratory & Departament of Computer Science. University of Maryland, 1993.

SHNEIDERMAN, B. **Designing the User Interface**, 3rd edition. Reading, MA: Addison Wesley, 1998.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 3ª edição. São Paulo: Makron Books, 1999.

SILVA S.; CATARCI T.; SCHIEL U. **"A Graphical Notebook" as Interaction Metaphor for Queryng Databases**. XIV Simpósio Brasileiro de banco de Dados, *Florianópolis*, 1999.

SOUZA, C. S. de. **The Semiotic Engineering of User Interface Languages**. In: Int. Man-Machine Studies, 39, 753-773, 1993.

SOUZA, C.S. de; MARTINS, I. H., **Uma Abordagem Semiótica na Utilização dos Recursos Visuais em Linguagens de Interface**, IHC'98, 38-47, 1998.

SOUZA, C. S. de; LEITE, J. C; PRATES, R. O; BARBOSA, D. J. **Projeto de Interfaces de Usuário: Perspectivas Cognitivas e Semióticas**. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 1999.

SPACCAPIETRA, S.; DENNEBOUT, Y; GENTILE, M.; FONTANA, E.; YANN, D.; AUDDINO, A.; ANDERSSON, M. **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.

SPACCAPIETRA, S.; PARENT, C.; SUNYE, M. S.; YETONGNON, K.; LEVA, A. di. **An Object + Relationship Paradigm for Database Applications**. Readings in Object-Oriented Systems, D. Rine (Ed.), IEEE Press, 1995.

SOWA, J. F. **Knowledge representation: logical, pjilosophical and computational foundations**. Brooks, Cole. USA, 2000.

VANDIVIER, S. E. **Discoverer 3.1 - Strategies for Proactive End User Layer Administration**. Disponível em http://www.avanco.com/ discover_strategies.htm, 2000. Acesso: 25 jul. 2000.

ZLOOF, M. M. **Query-by-Example: A Data Base Language**. IBM Systems Journal, Volume 16, Number 4, 324-343, 1977.

APÊNDICE A - LEMD DO AMBIENTE PROPOSTO

```
Command-message Etapa de Definição de Predicados

Command-message Construção da Consulta for Application-
Function Consulta
Join {
     Text "Efetuar a construção da consulta, selecionando os
atributos desejados na resposta e estabelecendo critérios.
Após, verifique a sentença na caixa 'Consulta em Elaboração'
e clique no botão 'Executar'"
     View "Visualização do Sub-esquema Selecionado:"
//Selecionado previamente pelo usuário
     Sequence {
          View "Selecionar Atributos ..."
          View 'Estabelecer Critérios ..."

          Command-message Sub-etapa: Escolher Atributos para a
     Exibição for Application-Function Seleção  //Ativa Seleção
     de Atributos
          Repeat {
               Sequence{
                  Text "Nesta fase, você deverá selecionar os
atributos que serão exibidos na resposta da consulta"
                  View "Tabelas Disponíveis"
                  Combine {
                           Select information-of Tabelas
                           View "Atributos Disponíveis"
                           Enter information of <atributos
disponíveis>
                           View "Atributos Selecionados"
                           Select{
                             Activate Adicionar_itens
Application-Function Seleção
                             Activate Adicionar_todos
Application-Function Seleção
                             Activate Retirar_itens
     Application-Function Seleção
                             Activate Retirar_todos
Application-Function Seleção
                           }
                         }
                  }
            Activate undo command-message        "Etapa
     Anterior"
```

```
            Activate start Application-Function "Ajuda"
            Activate start Application-Function "Confirmar"
        Text " Nenhum atributo selecionado!"
        Activate start Application-Function "OK"
    } // Fim da Função Ativa Seleção de Atributos

        Command-message Sub-etapa: Estabelecer Critérios for
    Application-Function Critérios
        Join {
            Text "Estabeleça as restrições de domínio
    desejadas (Expressão Relacional), comparando atributos
    entre si ou fornecendo um valor constante para comparação
    e clique em 'Inserir Expressão'"
            View "Expressão Relacional"
            View "Tabelas Disponíveis"
            Select Information-of Tabelas
            View "Atributos Disponíveis"
            Select information-of Atributos disponíveis
            View "Operador Relacional"
            Select information-of operador
            Select{
                View "Tabelas para Comparação"
                Select information-of Tabelas_Comparação
                View "Atributos para Comparação"
                Select information-of Atributos_Comparação
                View "Valor:"
                Enter information-of Valor
                }
            Active show command-message "Inserir Critério"
            Text "Se o número de expressões inseridas for
maior que 2 (dois), deve-se selecionar os critérios e aplicar
operador(es) lógico(s) para agrupar as condições a serem
aplicadas."
            View "Construção dos Critérios"
            Enter information-of Critérios
            View "Operador Lógico"
            Enter information-of Critérios
            Select information-of Operador
            Select information-of Critérios
            Activate start Application-Function Aplicar
        Operador
            Activate start Application-Function Excluir
        Critério
        }
        Text "Devem ser selecionados pelo menos dois
critérios"
        Activate start Application-Function "OK"
```

```
            Text "Nenhum critério selecionado"
            Activate start Application-Function "OK"
            Text "Operador lógico não selecionado"
            Activate start Application-Function "OK"
      }
      Activate Undo Application-Function "Etapa Anterior"
      Activate Start Application-Function "Ajuda"
      Activate Start Application-Function "Opções Avançadas"
      Activate Start Application-Function "Confirmar"

Command-message Opções Avançadas for Application-Function
Opções Avançadas
      Join{
            Select{
                  Join{
                        Text "Escolha operadores a serem também
aplicados à consulta."
                        View "Outras Opções:"
                        View "Aplicar Operador de Redução"
                        Enter information-of Redução
                        Text "Selecione o operador de redução se
                  quiser que a aplicação da condição à entidade
                  "filha" não interfira na exibição, no resultado,
                  das ocorrências da entidade "pai"."
                        View "Aplicar Operador de Simplificação"
                        Text "Selecione o operador de simplificação
                  se quiser que a subordinação entre as
                  ocorrências da entidade "pai" e da entidade
                  "filha" (com valores correspondentes) não seja
                  considerada no resultado."
                        Enter information-of Simplificação
                      }
                  }
            } // Fim Função Opções Avançadas

View "Consulta em Elaboração"
Enter information-of <feedback> //Dado pelas ações executadas
pelo usuário na interface
}
Activate Undo Application-Function "Etapa Anterior" //Seleção
de sub-esquema
Activate Start Application-Function "Executar"
Activate Start Application-Function "Ajuda"
Activate Close command-message "Construção da Consulta"

} // Fim da Etapa de Construção da Consulta → Início da Fase
de Exibição da Resposta
```

APÊNDICE B - STAG DO AMBIENTE DE INTERFACE PROPOSTO

Declaração de Tipos e Instâncias

.....

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

primitiva

primitiva-interação

primitiva-composta

plano-interação

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

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:raiz_sub-esquema, diagrama_sub-esquema

plano-interação: plano, janela_feedback

.....

1. Tipos de Operações

operações-edição

1. Instâncias de Operações

operações-edição: marcar, criar_predicado

.....

1. Tipos de Variáveis de Estado das Entidades

estado-seleção

1. Instâncias de Variáveis de Estado das Entidades

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: janela-rolagem, cursor-cruz, janela

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

cor: amarelo, preto, azul

texto: menu-hierárquico, rótulo

.....

1. Tipos do Conteúdo do Usuário

intenção

1. Instâncias do Conteúdo do Usuário

intenção: indicar, movimentar, ativar_janela, agrupar, ativar_menu, digitar, posicionar

.....

1. Tipos da Expressão do Usuário

sinal

1. Instâncias da Expressão do Usuário

sinal: clique-botão-esquerdo, pressão-e-arrasto, clique-duplo, clique + pressão-tecla <shift>, clique-botão-direito, pressão-tecla, parar-com-mouse

.....

Mapeamento

.....

1. Mapeamento de Tipos do Sistema

primitiva <=> forma

primitiva-interação <=> cor

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

primitiva-composta <=> diagrama

plano-interação <=> diagrama <> cor <> janela_feedback

.....

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 (azul) <> forma-composta (cursor-cruz)

estado-seleção (inativo) <=> nulo

operação-edição (criar_predicado) <=> forma-composta (janela) <> texto (rótulo)

.....

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-duplo)

intenção (agrupar) <=> sinal (clique + pressão-tecla <shift>)

intenção (ativar_menu) <=> sinal (clique-botão-direito + clique-botão-esquerdo)

intenção (digitar) <=>sinal (pressão-tecla)

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

.....

Dicionário

.....

1. Dicionário de Tarefas Básicas

selecionar-objeto [operação = indicar, critério = único]

selecionar-grupo [operação = agrupar, critério = grupo]

inserir-redução [operação = indicar, objeto = atributo]

inserir-simplificação [operação = indicar, objeto = atributo]

construir-predicado [operação = digitar, objeto = sub-esquema, critério = grupo]

mover-atributo [operação = movimentar, objeto = atributo, critério = único]

.....

Gramática

.....

1. Regras Gramaticais

R1. Selecionar-objeto [critério = único] -->

Primitiva ( )

Intenção (indicar) +

Estado-seleção (ativo)

R2. Selecionar-grupo [critério = grupo] -->

Primitiva ( )

Intenção (agrupar) +

Estado-seleção (ativo)

R3. Selecionar-janela [janela] -->

Primitiva ( )
Intenção (ativar_janela) +
Estado-seleção (ativo)

R4. Inserir-redução [critério = único] -->
Intenção (indicar) +
Estado-seleção (ativo)

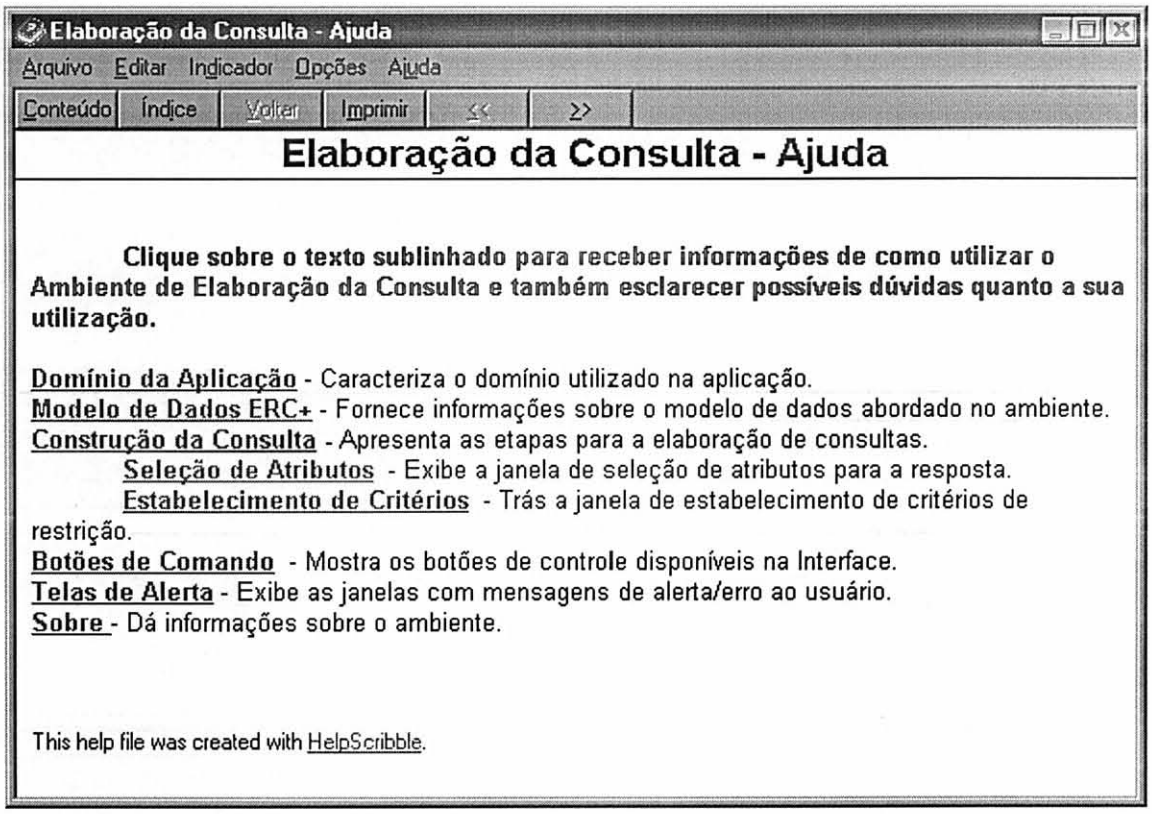
R5. Construir-predicado [critério = grupo] -->
Selecionar-objeto [critério = único, primitiva = atributo] +
Operação-edição (criar_predicado) +
Intenção (digitar) +
Inserir-redução
Inserir-simplificação

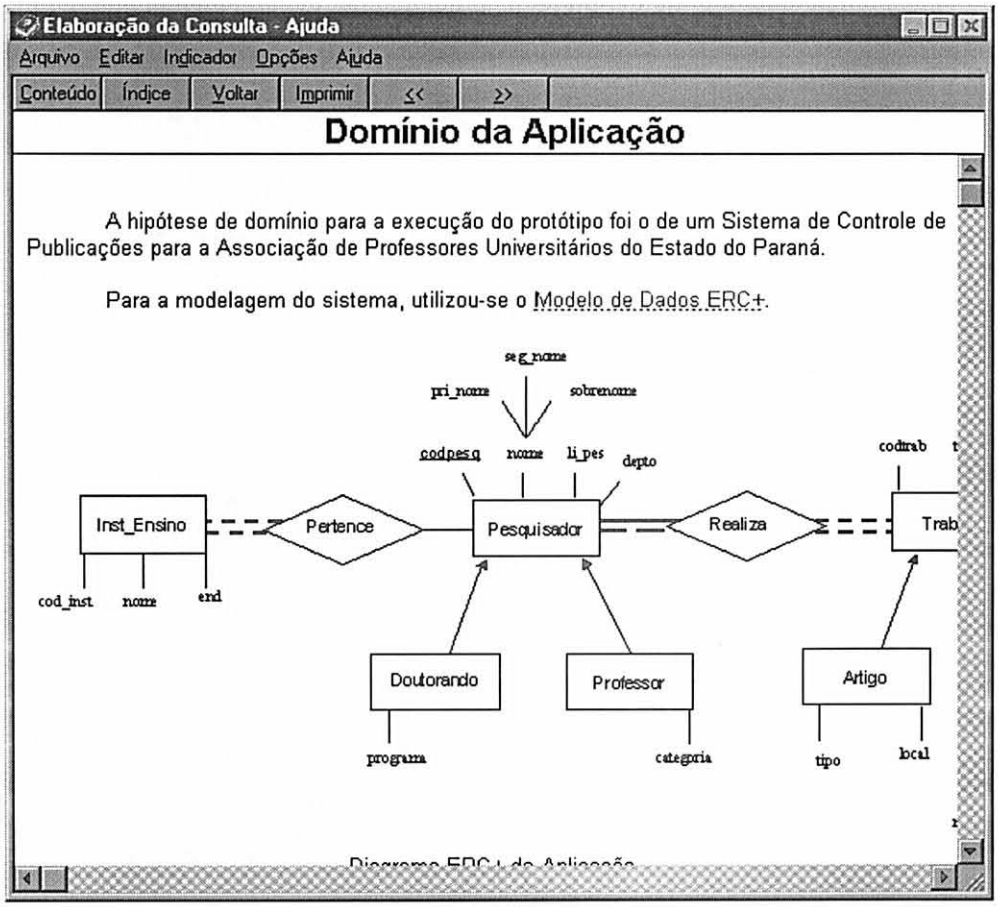
APÊNDICE C - *HELP* DO AMBIENTE DE ELABORAÇÃO DE CONSULTA

Ao clicar no botão ajuda, a ferramenta disponibiliza o sistema de *help* disponível, construído nos moldes *Windows* e de forma sistêmica, de forma a não despender esforço cognitivo do usuário em sua utilização.

Abaixo, a tela principal da Ajuda que além do menu padrão, disponível em qualquer *help* W*indows,* oferece vários *links* e menus *popup* para facilitar o aprendizado e a harmonia da interface.

A ajuda do sistema pode ser ativada a qualquer tempo e ainda, pode ser impressa. Adicionalmente aos botões Voltar, >> e <<, são oferecidas buscas por conteúdo ou índice.





Modelo ERC+

O modelo ERC+ (Entidade Relacionamento para Objetos Complexos, onde o + indica enriquecimento semântico) é um paradigma e metodologia claros baseados nas fundamentações de linguagem de programação favorecendo ao desenvolvimento de linguagens e interfaces orientadas ao usuário, pois objetiva uma maior descrição semântica.

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.

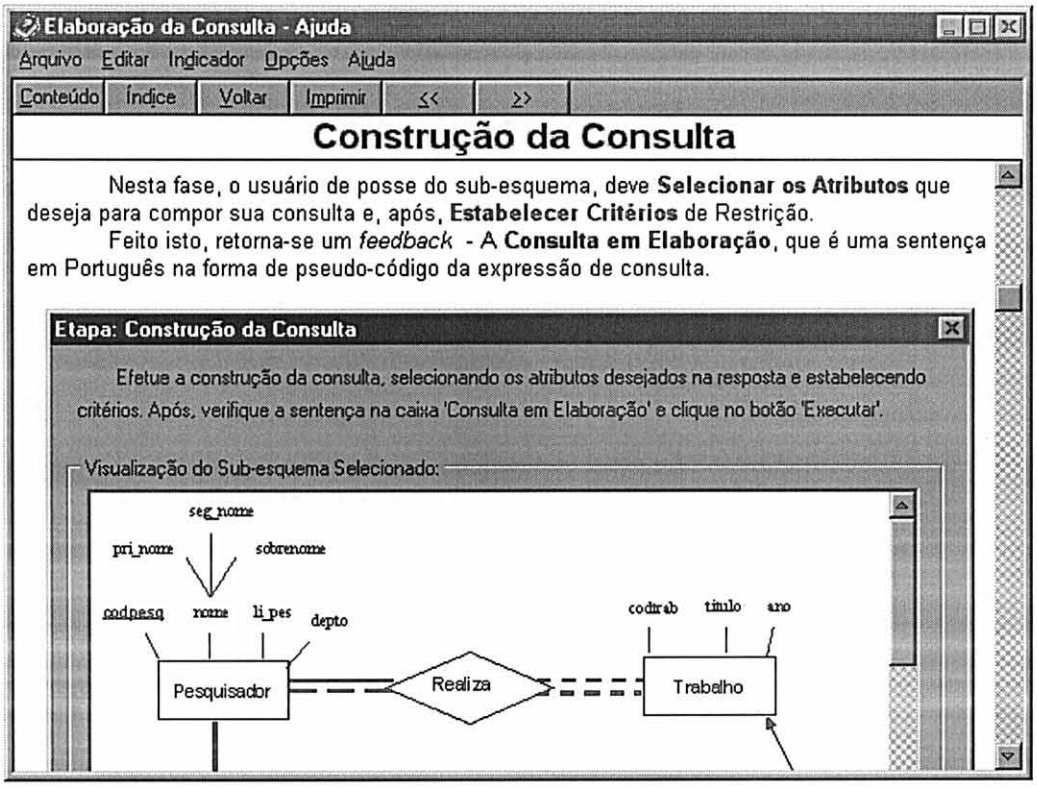
Um papel é associado a cada participação de uma entidade em um relacionamento. Este papel é caracterizado por seu mínimo e máximo de cardinalidades, especificadas como 0-1, 0-n, 1-1 ou 1-n de acordo com o vínculo da entidade para o relacionamento. Entidades e relacionamentos podem ter zero, um ou mais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

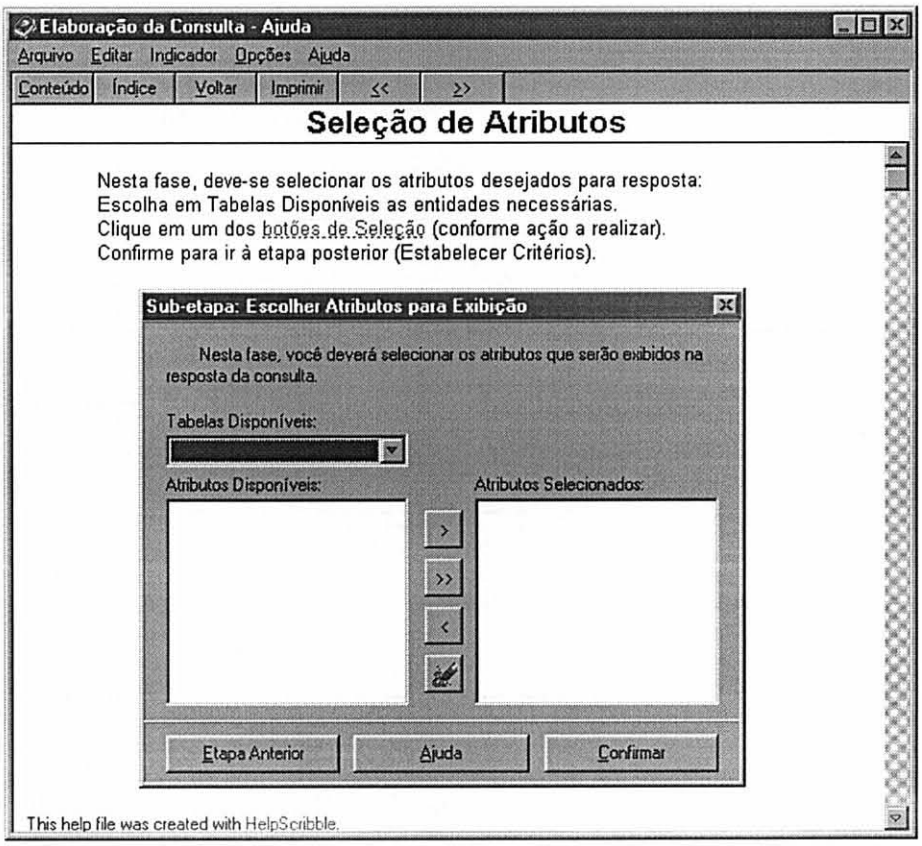
Atributos podem ser: obrigatórios ou opcionais, monovalorados ou multivalorados e simples ou complexos. Um atributo identificador indica que o valor deste atributo é distinto para cada instância desta entidade, sendo que o mesmo é utilizado como identificação da instância.

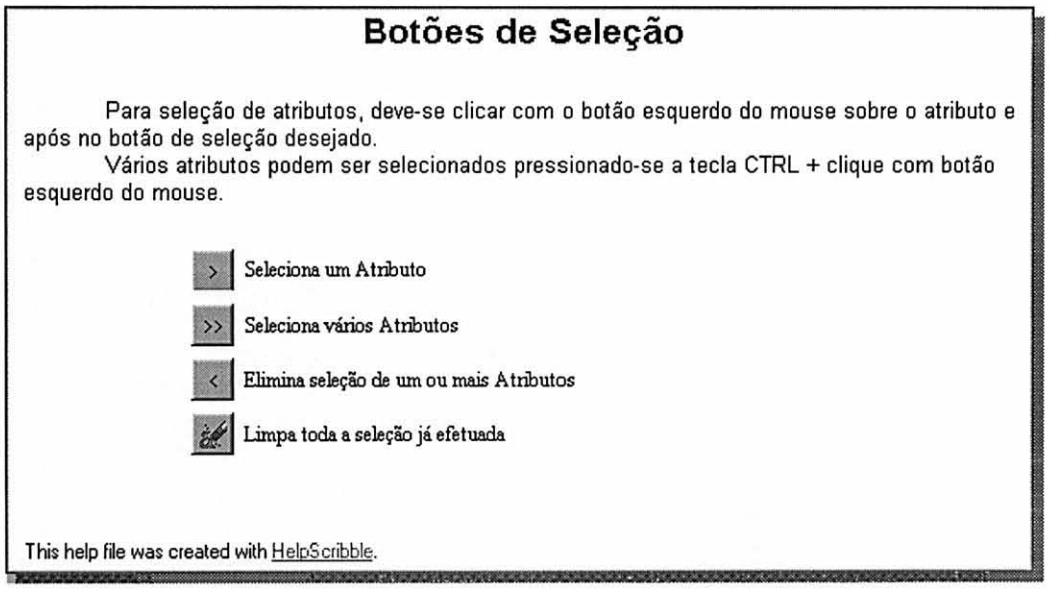
Duas generalizações são suportadas no modelo ERC+, "*is-a*" (identifica uma entidade genérica que possui as características principais de entidades específicas) e "*may-be-a*" (indica que a população do tipo entidade genérico não inclui o conjunto da população do tipo entidade específico, ou seja, uma consulta realizada no tipo entidade específico não abrange os dados do tipo entidade genérico).

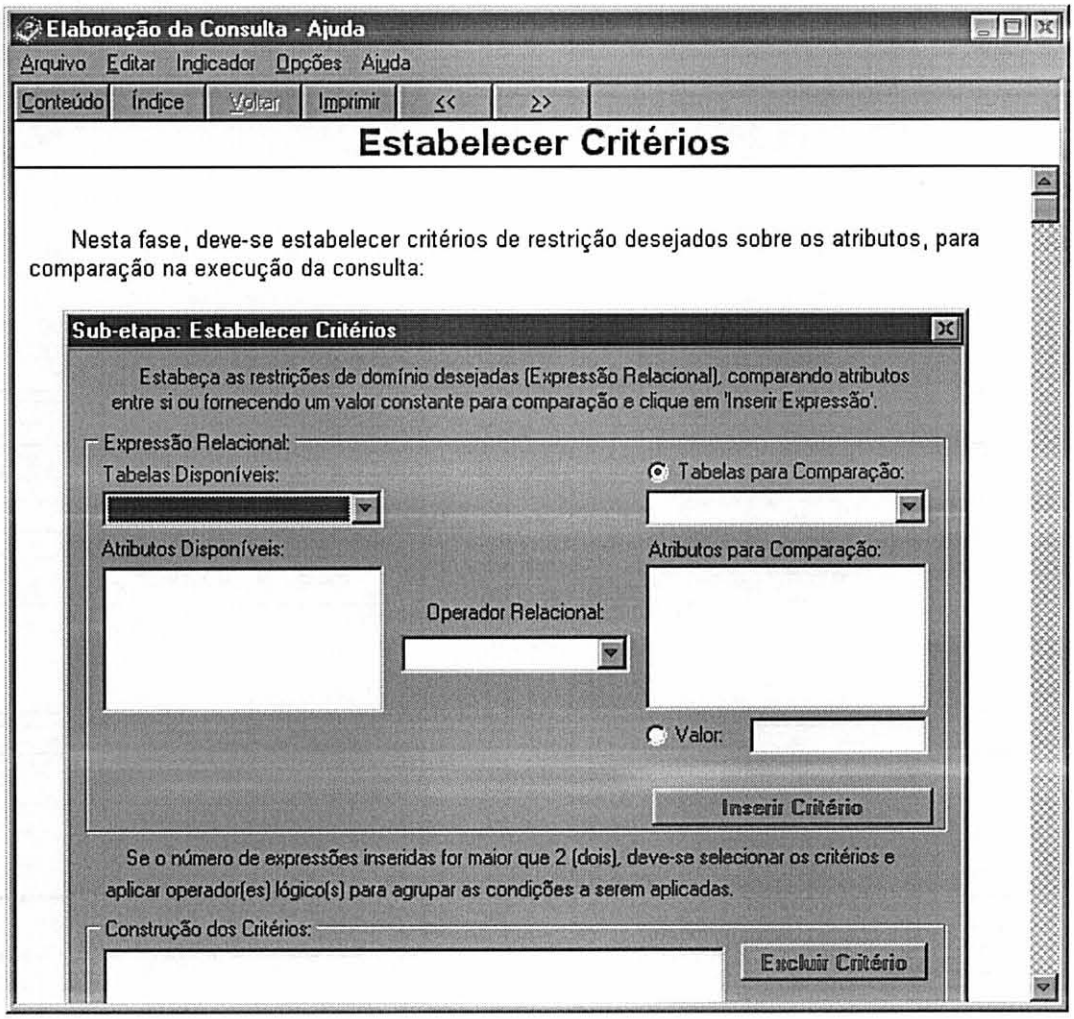
Quando existe a ocorrência de uma generalização, os atributos da entidade pai são comuns às entidades filhas, sendo que os atributos específicos de cada entidade são representados nas entidades filhas.

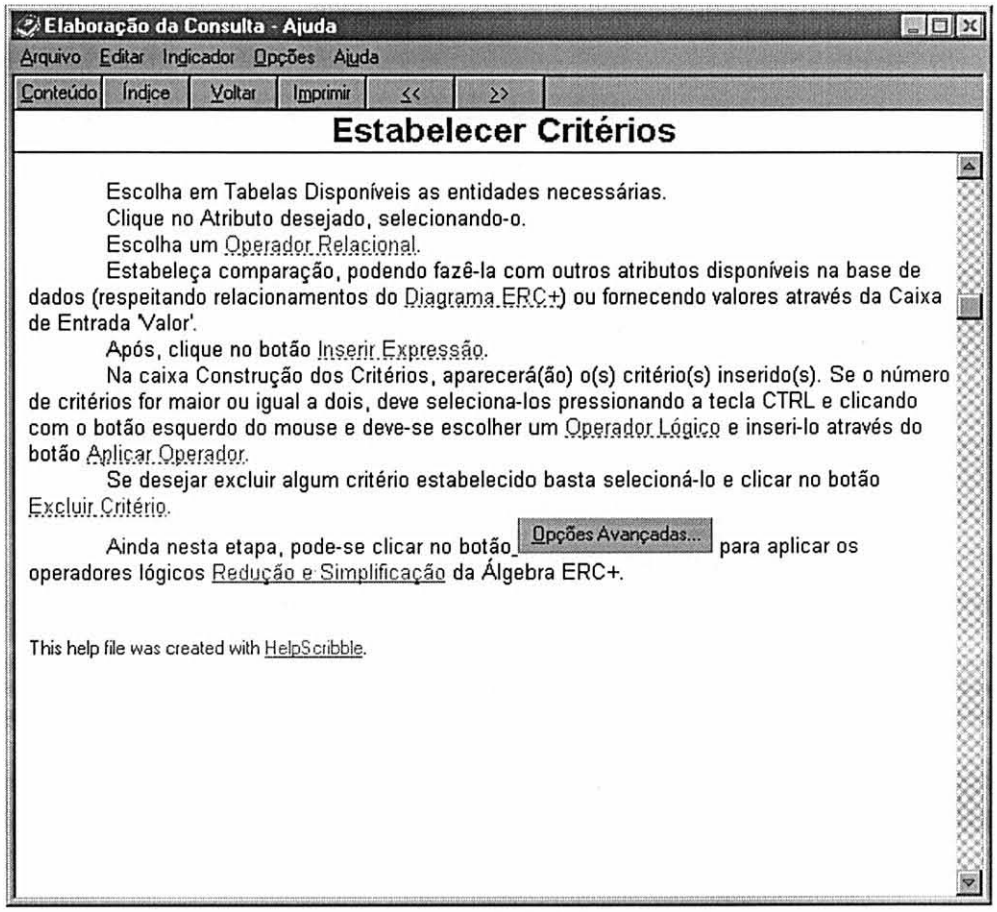
This help file was created with HelpScribble.

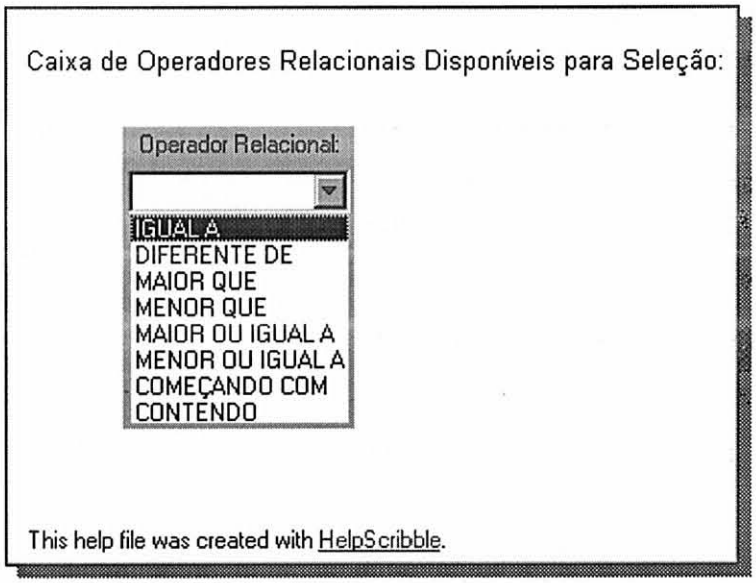

Elaboração da Consulta - Ajuda
Arquivo Editar Indicador Opções Ajuda
Conteúdo Índice Voltar Imprimir << >>
Construção da Consulta
Nesta fase, o usuário de posse do sub-esquema, deve Selecionar os Atributos que deseja para compor sua consulta e, após, Estabelecer Critérios de Restrição.
Feito isto, retorna-se um feedback - A Consulta em Elaboração, que é uma sentença em Português na forma de pseudo-código da expressão de consulta.
Etapa: Construção da Consulta
Efetue a construção da consulta, selecionando os atributos desejados na resposta e estabelecendo critérios. Após, verifique a sentença na caixa 'Consulta em Elaboração' e clique no botão 'Executar'.
Visualização do Sub-esquema Selecionado:
seg_nome
pri_nome
sobrenome
codpesq
nome
li_pes
depto
codtrab
titulo
ano
Pesquisador
Realiza
Trabalho


Elaboração da Consulta - Ajuda
Arquivo Editar Indicador Opções Ajuda
Conteúdo Índice Voltar Imprimir << >>
Seleção de Atributos
Nesta fase, deve-se selecionar os atributos desejados para resposta:
Escolha em Tabelas Disponíveis as entidades necessárias.
Clique em um dos botões de Seleção (conforme ação a realizar).
Confirme para ir à etapa posterior (Estabelecer Critérios).
Sub-etapa: Escolher Atributos para Exibição
Nesta fase, você deverá selecionar os atributos que serão exibidos na resposta da consulta.
Tabelas Disponíveis:
Atributos Disponíveis:
Atributos Selecionados:
Etapa Anterior
Ajuda
Confirmar
This help file was created with HelpScribble.


Botões de Seleção
Para seleção de atributos, deve-se clicar com o botão esquerdo do mouse sobre o atributo e após no botão de seleção desejado.
Vários atributos podem ser selecionados pressionado-se a tecla CTRL + clique com botão esquerdo do mouse.
Seleciona um Atributo
Seleciona vários Atributos
Elimina seleção de um ou mais Atributos
Limpa toda a seleção já efetuada
This help file was created with HelpScribble.


Elaboração da Consulta - Ajuda
Arquivo Editar Indicador Opções Ajuda
Conteúdo Índice Voltar Imprimir << >>
Estabelecer Critérios
Nesta fase, deve-se estabelecer critérios de restrição desejados sobre os atributos, para comparação na execução da consulta:
Sub-etapa: Estabelecer Critérios
Estabeça as restrições de domínio desejadas (Expressão Relacional), comparando atributos entre si ou fornecendo um valor constante para comparação e clique em 'Inserir Expressão'.
Expressão Relacional:
Tabelas Disponíveis:
Tabelas para Comparação:
Atributos Disponíveis:
Operador Relacional:
Atributos para Comparação:
Valor:
Inserir Critério
Se o número de expressões inseridas for maior que 2 (dois), deve-se selecionar os critérios e aplicar operador(es) lógico(s) para agrupar as condições a serem aplicadas.
Construção dos Critérios:
Excluir Critério


Elaboração da Consulta - Ajuda
Arquivo Editar Indicador Opções Ajuda
Conteúdo Índice Voltar Imprimir << >>
Estabelecer Critérios
Escolha em Tabelas Disponíveis as entidades necessárias.
Clique no Atributo desejado, selecionando-o.
Escolha um Operador Relacional.
Estabeleça comparação, podendo fazê-la com outros atributos disponíveis na base de dados (respeitando relacionamentos do Diagrama ERC+) ou fornecendo valores através da Caixa de Entrada 'Valor'.
Após, clique no botão Inserir Expressão.
Na caixa Construção dos Critérios, aparecerá(ão) o(s) critério(s) inserido(s). Se o número de critérios for maior ou igual a dois, deve seleciona-los pressionando a tecla CTRL e clicando com o botão esquerdo do mouse e deve-se escolher um Operador Lógico e inseri-lo através do botão Aplicar Operador.
Se desejar excluir algum critério estabelecido basta selecioná-lo e clicar no botão Excluir Critério.
Ainda nesta etapa, pode-se clicar no botão Opções Avançadas... para aplicar os operadores lógicos Redução e Simplificação da Álgebra ERC+.
This help file was created with HelpScribble.


Caixa de Operadores Relacionais Disponíveis para Seleção:
Operador Relacional:
IGUAL A
DIFERENTE DE
MAIOR QUE
MENOR QUE
MAIOR OU IGUAL A
MENOR OU IGUAL A
COMEÇANDO COM
CONTENDO
This help file was created with HelpScribble.

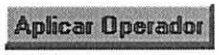
Ao clicar no botão Inserir Expressão, o usuário efetivamente estará criando um critério para comparação.

This help file was created with HelpScribble.

Caixa de Operadores Lógicos Disponíveis para Seleção:

This help file was created with HelpScribble.

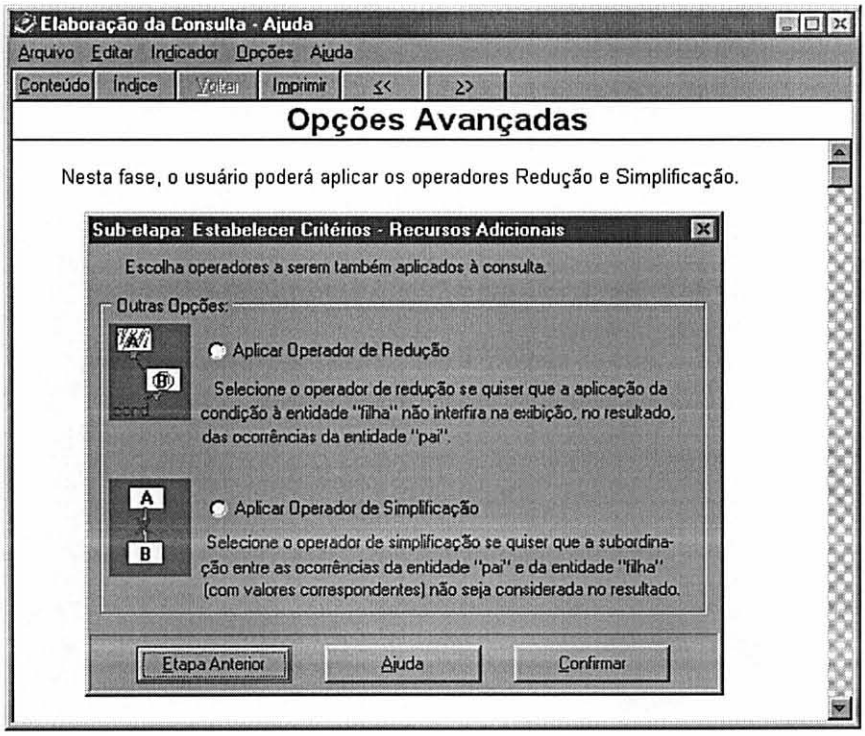
Ao clicar no botão Aplicar Operador, o usuário estará criando um critério de comparação agrupado. Este agrupamento é claramente denotado por parênteses.

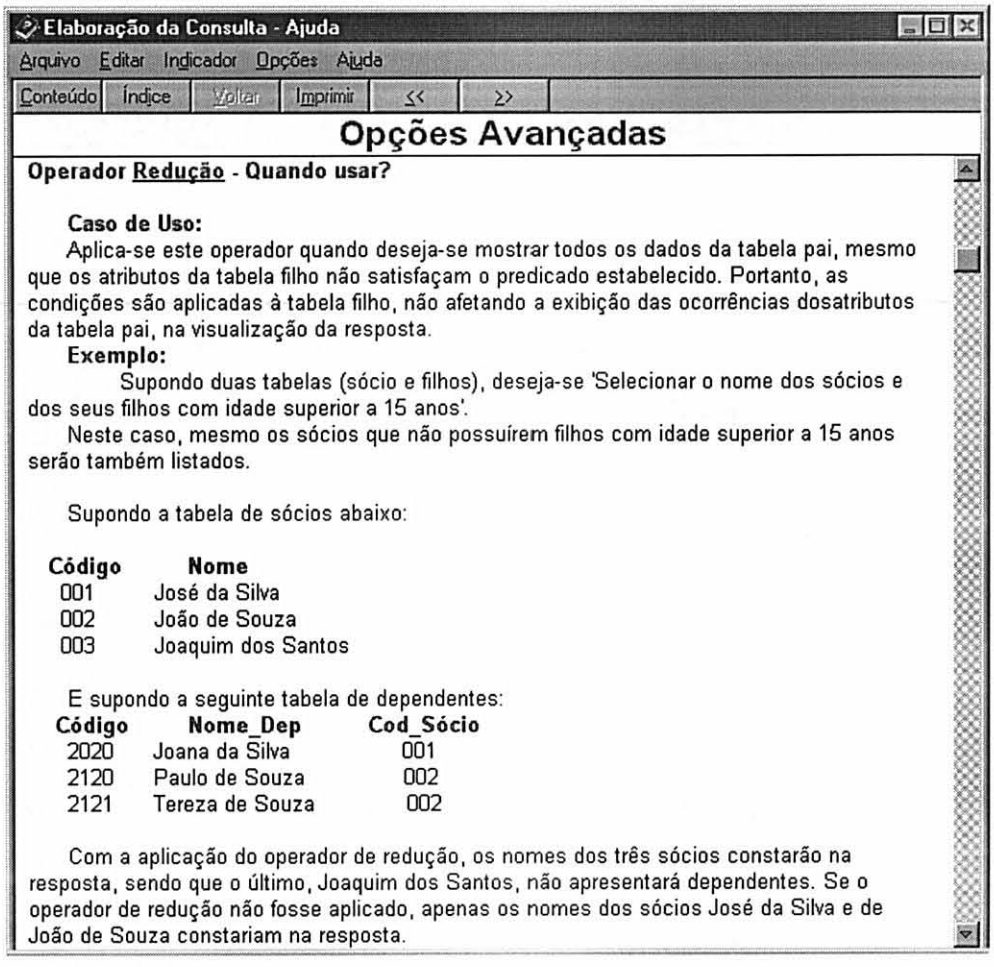


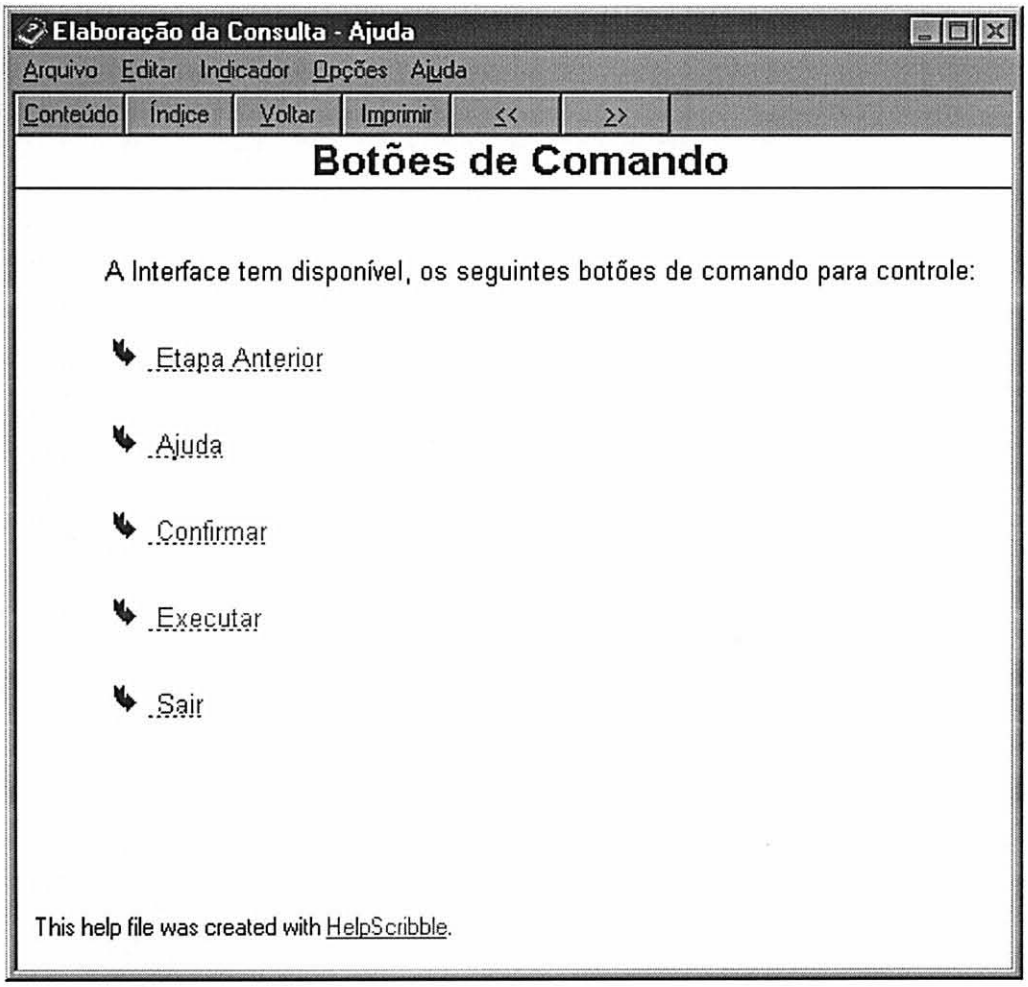
This help file was created with HelpScribble.

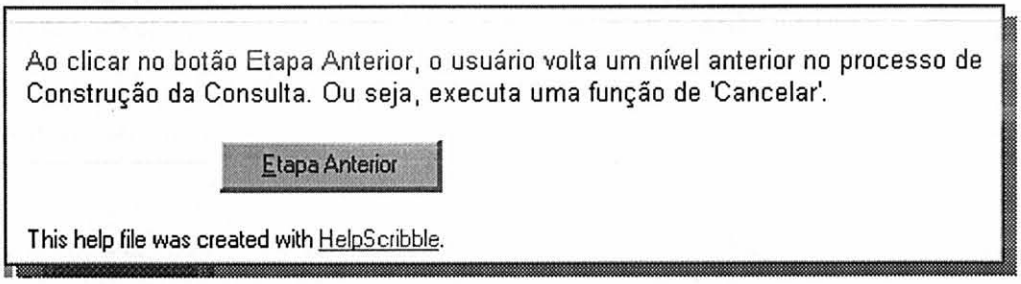
Ao clicar no botão Excluir Critério, o usuário estará excluindo um critério selecionado.

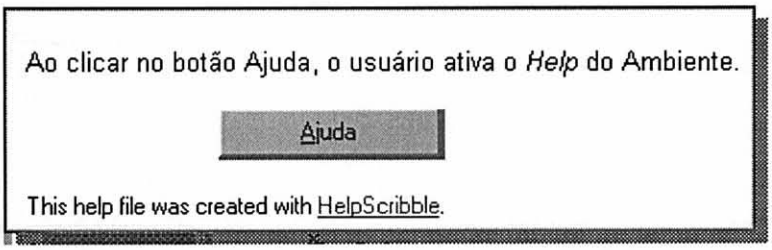
This help file was created with HelpScribble.


Elaboração da Consulta - Ajuda
Arquivo Editar Indicador Opções Ajuda
Conteúdo Índice Voltar Imprimir << >>
Opções Avançadas
Nesta fase, o usuário poderá aplicar os operadores Redução e Simplificação.
Sub-etapa: Estabelecer Critérios - Recursos Adicionais
Escolha operadores a serem também aplicados à consulta.
Outras Opções:
Aplicar Operador de Redução
Selecione o operador de redução se quiser que a aplicação da condição à entidade "filha" não interfira na exibição, no resultado, das ocorrências da entidade "pai".
Aplicar Operador de Simplificação
Selecione o operador de simplificação se quiser que a subordinação entre as ocorrências da entidade "pai" e da entidade "filha" (com valores correspondentes) não seja considerada no resultado.
Etapa Anterior
Ajuda
Confirmar


Elaboração da Consulta - Ajuda
Arquivo Editar Indicador Opções Ajuda
Conteúdo Índice Voltar Imprimir << >>
Opções Avançadas
Operador Redução - Quando usar?
Caso de Uso:
Aplica-se este operador quando deseja-se mostrar todos os dados da tabela pai, mesmo que os atributos da tabela filho não satisfaçam o predicado estabelecido. Portanto, as condições são aplicadas à tabela filho, não afetando a exibição das ocorrências dosatributos da tabela pai, na visualização da resposta.
Exemplo:
Supondo duas tabelas (sócio e filhos), deseja-se 'Selecionar o nome dos sócios e dos seus filhos com idade superior a 15 anos'.
Neste caso, mesmo os sócios que não possuírem filhos com idade superior a 15 anos serão também listados.
Supondo a tabela de sócios abaixo:
Código Nome
001 José da Silva
002 João de Souza
003 Joaquim dos Santos
E supondo a seguinte tabela de dependentes:
Código Nome_Dep Cod_Sócio
2020 Joana da Silva 001
2120 Paulo de Souza 002
2121 Tereza de Souza 002
Com a aplicação do operador de redução, os nomes dos três sócios constarão na resposta, sendo que o último, Joaquim dos Santos, não apresentará dependentes. Se o operador de redução não fosse aplicado, apenas os nomes dos sócios José da Silva e de João de Souza constariam na resposta.


Elaboração da Consulta - Ajuda
Arquivo Editar Indicador Opções Ajuda
Conteúdo Índice Voltar Imprimir << >>
Botões de Comando
A Interface tem disponível, os seguintes botões de comando para controle:
Etapa Anterior
Ajuda
Confirmar
Executar
Sair
This help file was created with HelpScribble.


Ao clicar no botão Etapa Anterior, o usuário volta um nível anterior no processo de Construção da Consulta. Ou seja, executa uma função de 'Cancelar'.
Etapa Anterior
This help file was created with HelpScribble.


Ao clicar no botão Ajuda, o usuário ativa o Help do Ambiente.
Ajuda
This help file was created with HelpScribble.

Ao clicar no botão Confirmar, o usuário estará confirmando a ação executada.

Confirmar

This help file was created with HelpScribble.

Ao clicar no botão Executar, o usuário estará passando à etapa de Visualização da Resposta.

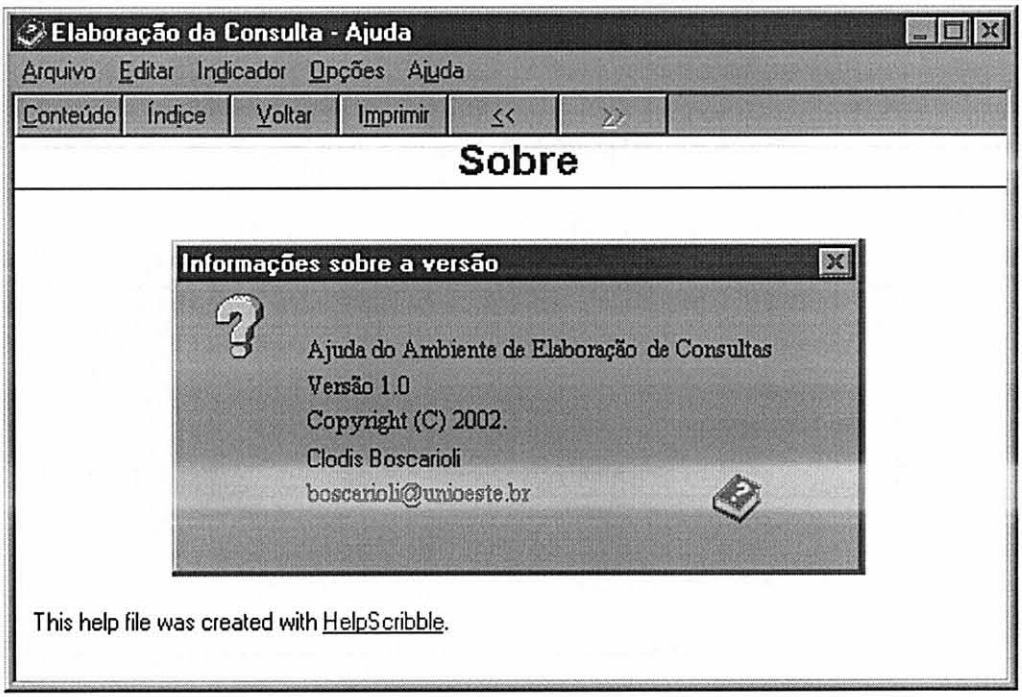
Executar

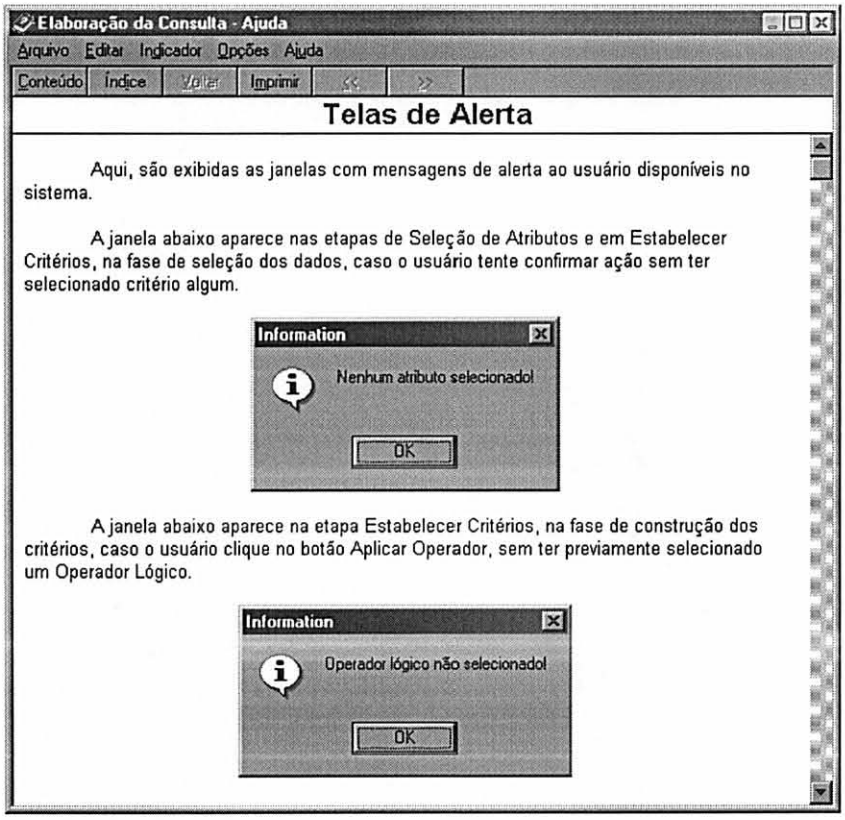
This help file was created with HelpScribble.

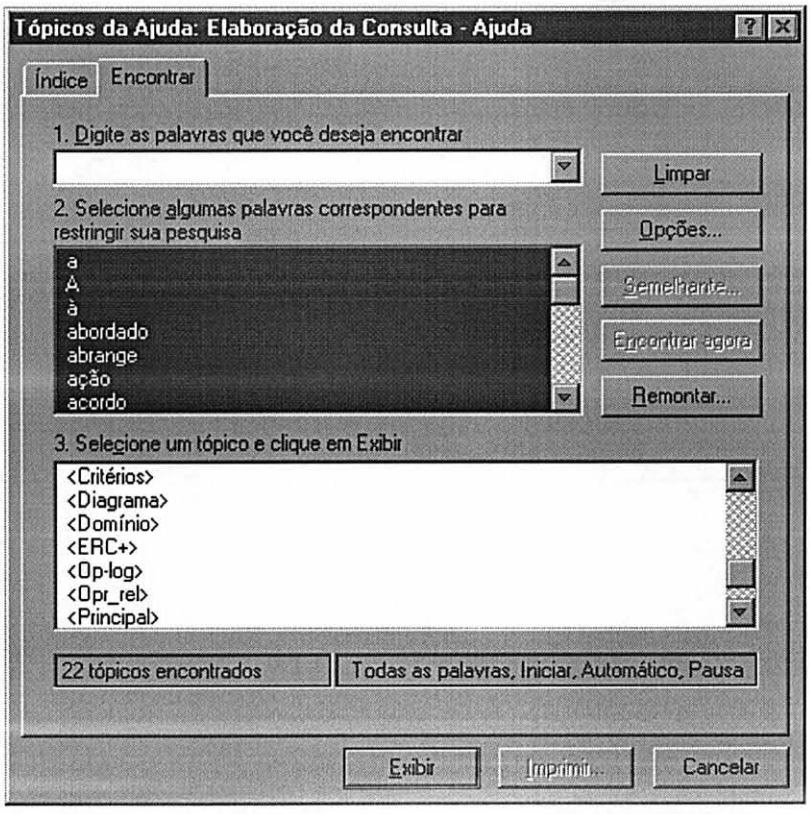
Ao clicar no botão Sair, o usuário estará fechando o Ambiente de Construção de Consultas, retornando ao Sistema Operacional.

Sair

This help file was created with HelpScribble.




Elaboração da Consulta - Ajuda
Arquivo Editar Indicador Opções Ajuda
Conteúdo Índice Voltar Imprimir
Telas de Alerta
Aqui, são exibidas as janelas com mensagens de alerta ao usuário disponíveis no sistema.
A janela abaixo aparece nas etapas de Seleção de Atributos e em Estabelecer Critérios, na fase de seleção dos dados, caso o usuário tente confirmar ação sem ter selecionado critério algum.
Information
Nenhum atributo selecionado!
OK
A janela abaixo aparece na etapa Estabelecer Critérios, na fase de construção dos critérios, caso o usuário clique no botão Aplicar Operador, sem ter previamente selecionado um Operador Lógico.
Information
Operador lógico não selecionado!
OK


Tópicos da Ajuda: Elaboração da Consulta - Ajuda
Índice
Encontrar
1. Digite as palavras que você deseja encontrar
Limpar
2. Selecione algumas palavras correspondentes para restringir sua pesquisa
Opções...
a
A
à
abordado
abrange
ação
acordo
Semelhante...
Encontrar agora
Remontar...
3. Selecione um tópico e clique em Exibir
<Critérios>
<Diagrama>
<Domínio>
<ERC+>
<Op-log>
<Opr_rel>
<Principal>
22 tópicos encontrados
Todas as palavras, Iniciar, Automático, Pausa
Exibir
Imprimir...
Cancelar

ANEXO A - OPERADORES DA ÁLGEBRA ERC+

Operador *selection* (Seleção)

É similar ao da álgebra relacional. A operação:

E = *select* [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 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.

Operador projection (Projeção)

É similar ao operador da álgebra relacional.

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 sub-atributos 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 (Redução)

É um operador novo, que complementa as funcionalidades oferecida pelos operadores *selection* e *projection* com respeito à meta de selecionar a informação desejada de um tipo entidade. Através da seleção e projeção permite ao usuário

descartar ocorrências ou atributos que não lhe interessem, a redução lhes 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.

Operador union (União)

Diferente do operador da álgebra relacional, o qual é baseado em valor, 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 *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 (Junção Relacional)

É um operador n-ário com um operando principal. Ele é usado para transformar um 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: $E = E_1$ ü $role_1$ *r-join* ($A : R, E_2$ ü $role_2$, ... , E_n ü $role_n$) onde $role_i$ é o papel executado pelo tipo entidade E_1 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 multivalorado denominado *A*, que é derivado dos esquemas de E_1, ..., 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, ..., E_n e ocorrências de valores de *R* que estão ligados a E_1 (se existir algum).

Operador i-join (Junção identidade)

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: $E = E_1$ *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 uma nome para o novo atributo complexo que será criado, E é um tipo entidade derivado, cujos oids são os mesmo 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 (Produto)

É usado colocar tipos entidades não relacionados dentro de uma entidade simples. Este resultado é similar ao operador de produto relacional NF2[4], 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 mesmo 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 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* (operador para troca de nomes) e *simplification* (Simplificação), os quais permitem adequar o esquema de uma entidade às regras do modelo ou da própria álgebra:

- operador *renaming* muda o nome de um atributo para preparar a fusão de atributos similares durando uma união.
- operador *simplification* exclui complexidade desnecessária na estrutura que pode ser construída por outros operadores, a projeção em particular. A simplificação exclui um nível na estrutura complexa quando um atributo

complexo tem somente um atributo componente (exceto se são ambos multivalorados).

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

O operador ***difference*** (Diferença ou -) pode ser definido através da composição apropriada do produto, seleção e projeção. Uma ***intersection*** (Interseção) 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* (junção externa) 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 *sel-r-join* é equivalente a uma expressão ERC+ feito sobre um *r-join* seguido por uma seleção.

Do mesmo modo, outro operador derivado, *v-join* (Junção por valor), 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.

---

[4] Modelo relacional aninhado que descreve estruturas que não estão na primeira forma normal.