SISTEMAS DE CONSULTAS VISUAIS PARA SGBD-OO_S COM INTERFACES INTELIGENTES: ARQUITETURA E IMPLEMENTAÇÃO

Sandra Martins Lino

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO

Aprovada por:

Prof. Geraldo Bonorino Xexéo, D.Sc.

(Presidente)

Profª. Marta Lima de Queirós Mattoso, D.Sc.

Profª. Maria Luiza Machado Campos, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

JANEIRO DE 1999

LINO, SANDRA MARTINS

Sistemas de Consultas Visuais para SGDB-OO_S com Interfaces Inteligentes: Arquitetura e Implementação [Rio de Janeiro] 1999

XII, 128 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1999)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Sistemas de Consulta

2. Interfaces Inteligentes

I. COPPE/UFRJ II. Título (série)

Aos meus pais e ao meu marido.

Agradecimentos

Ao meu orientador Geraldo Bonorino Xexéo pela sua orientação e dedicação durante a realização deste trabalho.

Aos professores do PESC pelas aulas maravilhosas, pelas opiniões e incentivos, em especial aos professores Marta Mattoso por seu exemplo, sua dedicação e simplicidade e Jano de Souza pela sua visão sempre além.

Aos meus pais que me possibilitaram estudar e chegar até aqui, provendo as condições psicológicas e financeiras para realizar este trabalho.

Ao meu marido Claudio que sempre me apoiou e incentivou em todas as minhas decisões, concordando ou não com elas, e que foi privado de muitas coisas em função de minha dedicação a este trabalho.

Aos meus colegas do PESC, que cada qual a sua maneira me animou e incentivou durante a realização deste trabalho, em especial a amiga Fernanda que me ajudou muito com suas opiniões e conselhos.

A CAPES pelo suporte financeiro deste trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SISTEMAS DE CONSULTAS VISUAIS PARA SGBD-OO_S COM INTERFACES INTELIGENTES: ARQUITETURA E IMPLEMENTAÇÃO

Sandra Martins Lino

Janeiro/1999

Orientador: Geraldo Bonorino Xexéo.
Programa: Engenharia de Sistemas e Computação

Usuários inexperientes tem dificuldades em criar, consultar e manter informações em banco de dados. Atualmente, essas operações são bastante frequentes e necessárias, entretanto nem todos os usuários possuem o conhecimento necessário para realizá-las, tanto no nível sintático quanto semântico.

Para minimizar estas dificuldades é necessário definir novos métodos de interação, baseados no gerenciamento indireto. Esses métodos são implementados por programas inteligentes que atuam como intermediários na comunicação entre os usuários e outros programas, obtendo-se um processo cooperativo de iniciação, manutenção e controle das tarefas realizadas no computador.

Esta tese propõe uma arquitetura para o desenvolvimento de sistemas de consultas visuais para banco de dados orientado a objetos e apresenta a implementação de um protótipo que valida a arquitetura. Esta arquitetura permite a realização do gerenciamento indireto utilizando os conceitos de interfaces inteligentes.

A implementação apresentada utiliza os ícones como representação visual para apresentar o domínio de interesse da consulta e criar as sentenças que serão transformadas em consultas OQL.

A técnica de interfaces inteligentes utilizada foi a de interfaces adaptativas. Esta técnica baseia-se na modificação gradual da interface do sistema de acordo com a evolução do conhecimento do usuário sobre a tarefa executada e o modelo de dados consultado. Tanto o volume de informações quanto os elementos visuais apresentados na interface podem ser modificados cada vez que o usuário realiza uma consulta no sistema.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

VISUAL QUERY SYSTEMS FOR OO DBMS WITH INTELLIGENT INTERFACES: ARCHITECTURE AND IMPLEMENTATION

Sandra Martins Lino

Janeiro/1999

Advisor: Geraldo Bonorino Xexéo
Department: Computer Science and Systems Engineering

It is very difficult for inexperient database users to create, query and mantain database information. This operations are very frequent and necessary nowadays but not all users have sufficient knowledge for these tasks, neither in the sintatic nor in the semantic level.

To minimize this problem, it is necessary to define new interaction methods, based on indirect management. These methods are implemented by intelligent programs that intermediate the communication between the user and other programs, creating a cooperative process where the inicialization, mantainance and control of computer tasks can take place.

This work proposes an architecture to support the development of visual query systems for object oriented databases and present an implementation prototype that validates the proposed architecture. This architecture perform the indirect management using intelligent interface concepts.

The implementation uses the iconic visual representation to display the query domain of interest and to create sentences that will be transformed in OQL queries.

We have used adaptative interfaces as our intelligent interface technique. This technique gradually modifie the system interface based on the user evolution of the knowledge about the task being executed and the data model queried. Both the information contents and the displayed visual elements of the interface can be modified each time the user queries the database from the system.

Índice do Texto

Índice de Figuras

Índice de Tabelas

Capítulo 1 -Introdução

Esta tese tem como objetivo principal propor uma arquitetura para o desenvolvimento de sistemas de consulta para banco de dados orientado a objetos, que implementem mecanismos de cooperação homem-máquina.

O conceito de cooperação foi inicialmente inserido na computação para reproduzir os processos cooperativos do mundo não computacional, caracterizados, principalmente, pela grande interação entre pessoas de um grupo para realização de uma única tarefa. Para isso, foram desenvolvidos diversos sistemas com mecanismos de troca de informações, conhecimento e controle entre membros de grupos de pessoas que cooperem para realização de uma tarefa.

Este conceito evoluiu e, atualmente, adquiriu novas vertentes: não só os indivíduos podem cooperar mas os computadores, na forma de seus sistemas, podem cooperar com seus usuários e também entre si.

O principal fator de impulsão deste novo modelo de cooperação é o aumento do número de novos usuários, normalmente inexperientes, que utilizam os computadores como uma ferramenta de auxílio na realização de suas tarefas. Para esses usuários, os computadores e seus sistemas compõem os meios de realização de tarefas, que compõem seu objetivo real. Desta forma, estes sistemas não podem requisitar do usuário grandes intervalos de tempo, tanto para a tomada de decisões, quanto no processo de aprendizado do *software* utilizado.

1.1 Motivação

A motivação maior para esta pesquisa é a crescente utilização dos sistemas de gerência de banco de dados por usuários considerados não especialistas na área de informática, especificamente na área de banco de dados. A manutenção de informações

armazenadas em um banco de dados e a realização de consultas vêm se tornando uma constante entre as mais diversas categorias de pessoas. Muitas dessas informações e consultas são utilizadas para tomada de decisões pessoais e em empresas.

Desde o gerente de uma agência bancária, com formação administrativa, até uma dona de casa, que pode ou não ter uma formação profissional, todos possuem informações que gostariam de armazenar para posteriormente consultar. Em alguns casos, esses usuários esbarram na dificuldade de efetivar este processo de criação, consulta e manutenção por não terem o conhecimento necessário para realizar esta tarefa.

Os sistemas de consultas visuais retratados na literatura buscam amenizar as dificuldades de interação, utilizando paradigmas visuais como forma de apresentação e manipulação dos dados. Entretanto, como foi ressaltado por CATARCI *et al.*(1990), esses sistemas facilitam a construção de consultas para determinados tipos de usuários, mas ainda não são eficientes para todos os tipos. Os usuários com pouco conhecimento na área de banco de dados e no modelo de dados consultado ainda não são atendidos pela maioria desses sistemas.

Segundo MAES(1994), as principais dificuldades de interação com os programas de computador ocorrem principalmente em virtude da necessidade de manipulação direta, que requer do usuário a iniciação das tarefas e o seu acompanhamento, verificando as novas ocorrências. Essa forma de interação é viável para realização de atividades simples, como imprimir arquivos e executar programas, tornando-se bastante difícil de ser utilizada, por exemplo, na busca de informações em grandes redes de computadores ou na consulta em bancos de dados orientado a objetos.

Na tentativa de acabar com essas dificuldades, uma nova forma de interação vem sendo desenvolvida baseada no gerenciamento indireto, onde programas inteligentes atuam como intermediários entre os usuários e outros programas, possibilitando um processo cooperativo de iniciação, manutenção e controle das tarefas realizadas no computador.

1.2 Objetivos da tese

O objetivo principal desta tese é propor uma arquitetura para o desenvolvimento de sistemas de consulta, que incorpore mecanismos de gerenciamento indireto para garantir uma cooperação da máquina, sistema, com o usuário. Outro objetivo desta tese

é desenvolver um sistema de acordo com esta arquitetura, para que se possa verificar como o gerenciamento indireto pode ser utilizado nos sistemas de consulta.

O gerenciamento indireto, proposto nesta tese para os sistemas de consultas visuais, baseia-se na construção de sistemas com interfaces adaptativas, ou seja, interfaces que se alteram, de acordo com a evolução do conhecimento do usuário sobre a tarefa a ser executada. Desta forma, estão sendo unidas duas tecnologias: sistemas de consulta e de interfaces inteligentes.

Outra mecanismo de gerenciamento indireto utilizado é a realização de processos de relaxamento de consultas, ou seja, modificação, pelo sistema, das consultas construídas pelo usuário, para garantir a existência de resultados.

1.3 Organização da tese

Esta tese está dividida em seis capítulos. O primeiro capítulo é esta introdução. O segundo capítulo apresenta o estágio atual do processo de realização de consultas, que são os sistemas de consultas visuais. O terceiro capítulo trata das interfaces adaptativas, destacando alguns sistemas já implementados. O quarto capítulo apresenta uma arquitetura para o desenvolvimento de sistemas de consultas visuais com interfaces inteligentes. O capítulo seguinte apresenta uma implementação desta arquitetura, destacando principalmente o aspecto adaptativo das interfaces construídas, e finalmente no capítulo seis apresentamos algumas conclusões obtidas com a realização deste trabalho.

Capítulo 2 -Sistemas de Consultas Visuais

Os sistemas de consultas visuais destinam-se à recuperação de informações em banco de dados utilizando representações visuais, tais como formulários, diagramas e ícones, como forma de apresentação do domínio de interesse, bem como para construção das consultas. Esses sistemas se distinguem dos outros pela utilização de interfaces elaboradas que procuram auxiliar o usuário na execução de sua tarefa de manipulação do banco de dados.

A construção dessas interfaces elaboradas foi possível com o desenvolvimento dos mecanismos de interação gráfica, caracterizados principalmente pela possibilidade de manipulação direta dos objetos apresentados na tela e a utilização dos conceitos de janelas, ícones e menus.

CATARCI *et al.*(1991) apresentam as principais vantagens obtidas com a utilização de interfaces elaboradas nos sistemas de consultas visuais: a proximidade entre o modelo do usuário e o modelo apresentado; o usuário torna-se independente da linguagem de consulta; facilidade de aprendizado dos mecanismos de interação; melhora da eficiência por todos os tipos de usuário e a redução dos erros obtidos durante a realização das consultas.

Neste capítulo serão apresentados alguns sistemas de consultas visuais descritos na literatura, considerando sempre o modelo de dados utilizado, a representação visual adotada, a interface desenvolvida e a forma de criação das consultas.

As linguagens descritas neste capítulo destinam-se principalmente aos usuários finais, que muitas vezes desejam ter acesso às bases de dados sem utilizar aplicações construídas á priori.

Desta forma, o capítulo encontra-se dividido em 3 seções. A primeira apresenta uma introdução e alguns fatores interessantes a serem considerados durante a análise

dos sistemas de consultas visuais. A segunda seção apresenta sistemas e linguagens de consultas visuais e finalmente na última seção são apresentadas algumas conclusões sobre os sistemas apresentados e duas tabelas comparativas para avaliação dos sistemas descritos.

2.1 Introdução

Nesta seção vamos apresentar e explicar alguns critérios de avaliação dos sistemas de consultas visuais, para facilitar o entendimento e a comparação dos sistemas descritos neste capítulo.

CATARCI *et al.*(1991) apresentam uma classificação dos sistemas de consultas visuais para bancos de dados textuais, baseada em dois critérios:

- *a representação visual utilizada para a construção da consulta e apresentação dos resultados;*
- *as estratégias de interação.*

Esses critérios estão relacionados à comunidade de usuários dos sistemas. Cada tipo de representação visual e de estratégia de interação é mais adequado a um determinado tipo de usuário. É necessário realizar avaliações prévias sobre o público alvo do sistema para determinar a melhor representação visual a ser utilizada, se há necessidade de aplicar estratégias para o entendimento do domínio de interesse e de qual técnica de formulação de consultas deve-se adotar para o público alvo do sistema.

Além destes critérios, utilizamos também alguns outros relacionados às funções fornecidas pelo sistema e sua relação com o usuário que são :

- *possibilidade de armazenamento e reutilização de consultas*
- *adaptação do sistema ao usuário*

2.1.1 Representação Visual

As representações visuais consistem em modos de apresentar as informações de forma não textual, utilizando algum conjunto de símbolos gráficos. Os modos utilizados nos sistemas já desenvolvidos e descritos na literatura são: os formulários, diagramas, ícones e híbridos. A seguir, iremos descrever essas representações e apresentar exemplos que mostrem como as representações podem ser utilizadas neste domínio de aplicação.

Os formulários foram a primeira forma de implementação de uma visualização bidimensional, através do agrupamento de objetos que possuem uma mesma estrutura. Esses formulários podem ser apresentados em formato de tabelas ou no formato de perguntas e respostas.

Sua utilização é bastante comum em sistemas com modelo de dados relacional, como o descrito por ELMASRI e NAVATHE(1994). A Figura 2.1 apresenta um exemplo retirado deste trabalho.

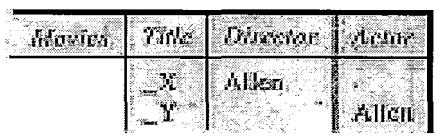


Figura 2.1 - Exemplo de formulário(ELMASRI e NAVATHE, 1994).

Os diagramas são representações gráficas que mostram de forma explícita os relacionamentos existentes entre os dados. Normalmente, utilizam componentes visuais bastante comuns para a construção dos diagramas como pontos e figuras geométricas. Em alguns casos, uma terceira dimensão, como cor e textura dos objetos visuais, é incluída para aumentar a capacidade de expressão. Outra possibilidade adotada, pelos sistemas já desenvolvidos, é a associação entre um objeto do diagrama e uma janela, que permite descrições mais detalhadas dos objetos visualizados no diagrama.

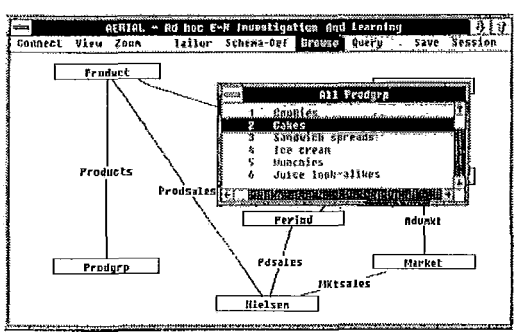


Figura 2.2 - Exemplo de diagrama que utiliza retângulos como forma geométrica e janelas auxiliares (BURNS et al., 1993).

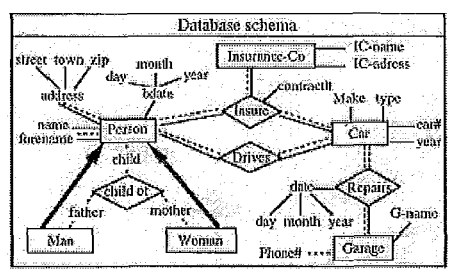


Figura 2.3 - Exemplo de diagrama que utiliza várias formas geométicas (AUDDINO *et al.*, 1992).

Através da Figura 2.2 e da Figura 2.3 apresentadas anteriormente, podemos perceber a existência de dois tipos de diagramas, com formas gráficas distintas para representação do modelo de dados da base a ser consultada.

Em sua grande maioria, estes diagramas são utilizados para navegação e em alguns casos também para construção de consultas.

Os ícones permitem a representação de um objeto ou conceito, através de imagens de objetos reais, por analogia ou por convenção. As consultas são formuladas através da combinação desses ícones, de acordo com uma sintaxe espacial. De acordo com CATARCI *et al.*(1991), os sistemas que utilizam esta forma de representação possuem interfaces mais amigáveis do que os sistemas que utilizam as formas descritas anteriormente, e o esquema do banco de dados não é apresentado. Esse tipo de sistema é mais aplicável a usuários inexperientes na tarefa. A Figura 2.4 representa uma consulta obtida em (TSUDA *et al.*, 1990), caracterizada pela composição de três ícones.

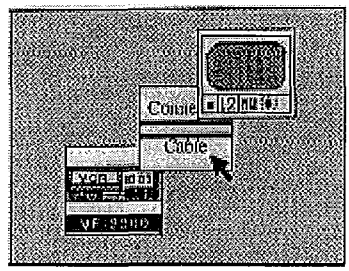


Figura 2.4 - Exemplo de ícones organizados para descrever uma consulta (TSUDA *et al.*, 1990).

Outra forma de apresentação utilizada é baseada na combinação de representações visuais, criando os chamados sistemas híbridos. Essas combinações podem ocorrer através de representações alternativas do banco de dados e das consultas, ou por uma combinação das representações descritas anteriormente, gerando novas formas visuais.

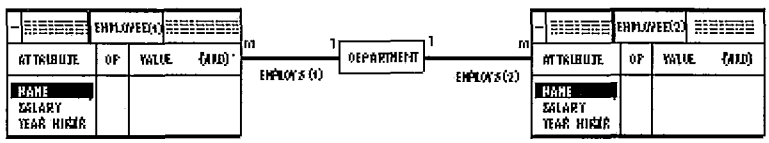


Figura 2.5 - Exemplo de representação híbrida, utilizando diagrama e formulários. (SOCKUT *et al.*, 1993).

Existem descritos na literatura protótipos de sistemas que utilizam representações visuais híbridas nas seguintes combinações: formulários & diagramas, diagramas & ícones e formulários, diagramas & ícones. A Figura 2.5 apresenta uma exemplo desta forma de representação visual.

2.1.2 Estratégias de Interação

De acordo com CATARCI *et al.*(1991), o processo de recuperação de informações de um banco de dados pode ser dividido em duas atividades básicas :

- *compreender o domínio de interesse da consulta;*
- *formulação da consulta.*

O entendimento do domínio de interesse pelo usuário é uma tarefa que anteriormente não era considerada pelos sistemas de consultas. Esta tarefa tem sua complexidade diretamente relacionada com o tamanho do esquema da base de dados e a carga do sistema.

Este processo de entendimento do modelo pode ser bastante lento, dependendo do conhecimento que o usuário já possui sobre o modelo. As estratégias desenvolvidas buscam garantir a construção gradual do modelo mental, com as informações por ele já capturadas sobre o modelo de dados analisado. Todas as novas informações capturadas geram uma evolução no modelo mental buscando torná-lo próximo da realidade.

Existem três estratégias que podem ser adotadas, a fim de, facilitar esta tarefa:

- *Refinamento de cima para baixo*

  Nesta estratégia, os aspectos mais gerais são apresentados antes dos detalhes mais específicos. Sua implementação pode ser através de refinamentos interativos sobre os esquemas, aproximação seletiva e aproximação hierárquica.

- *Navegação*

  Esta técnica consiste no estudo de elemento a elemento de um diagrama de entidades. Cada elemento selecionado é verificado, as informações úteis são adquiridas pelo usuário e o processo prossegue pelos elementos vizinhos.

  Esta navegação pode ser feita na intenção do banco de dados, ou seja, sobre o esquema, ou na sua extensão, que ocorre nos dados armazenados na base ou de ambas as formas.

- *Simplificação do esquema*

  Nesta estratégia uma visão do esquema é construída pelo usuário, através de agregações e transformações.

Dentre essas técnicas de conhecimento do domínio de interesse, a mais utilizada é a de navegação por diagramas, utilizada em (BURNS *et al.*, 1993), onde a navegação ocorre na parte intencional do banco de dados com possibilidade de visualização das extensões; (CAREY *et al.*, 1996) possibilita navegação pela extensão e (AUDDINO *et al.*, 1992) com navegação intencional. O sistema descrito por SANTUCCI e SOTTILE(1990) permite os refinamentos de cima para baixo bem como a simplificação do esquema, criando visões próprias do usuário.

A outra tarefa relevante ao processo de interação do sistema com o usuário está relacionada ao processo de formulação das consultas. Este processo está intimamente ligado à representação visual escolhida para o sistema. CATARCI *et al.* (1991) faz uma descrição detalhada das técnicas de formulação de consultas, classificando diversos sistemas. A seguir faremos apenas uma breve descrição das técnicas apresentadas.

➲ *Navegação no esquema*

Esta estratégia consiste em fixar-se um conceito ou um grupo de conceitos de interesse e a partir deste alcançar os demais conceitos envolvidos na consulta para o estabelecimento de suas restrições.

Essa navegação pode ser realizada através de um caminho conectado arbitrário, por um caminho hierárquico conectado ou por um caminho não conectado.

Alguns sistemas (AGRAWAL *et al.*, 1990), (CHANG *et al.*, 1996), (CAREY *et al.*, 1996) e (OLIVEIRA, 1992) implementam o conceito de sincronismo associado a navegação para consulta. A funcionalidade deste sincronismo determina que sempre que um objeto seja substituído por outro na apresentação, os objetos a ele relacionados, que estejam sendo apresentados, serão substituídos pelos objetos correspondentes ao objeto apresentado no momento. Dessa forma, as informações apresentadas na interface são sempre consistentes, possibilitando apresentar mais de um objeto por vez.

Para exemplificar, suponha a existência de um relacionamento ***Capital*** entre os objetos da classe ***País*** e ***Cidade*** e dois objetos da classe ***País*** representando ***Brasil*** e ***Portugal*** e dois da classe ***Cidade*** representando ***Brasília*** e ***Lisboa***.

Se o objeto correspondente ao ***Brasil*** estiver sendo apresentado bem como o relacionamento ***Capital***, então o objeto correspondente à cidade de ***Brasília*** estará sendo apresentado. Se ocorrer navegação nos objetos da classe ***País***, e o objeto correspondente à ***Portugal*** for selecionado, então automaticamente o objeto do relacionamento ***Capital*** a ser apresentado será o correspondente à cidade de ***Lisboa.***

- *Subconsultas*

  Nesta estratégia, a consulta é formulada através da composição de resultados parciais que podem ser conceitos, como ocorre nas linguagens baseadas em ícones, ou consultas previamente armazenadas.

- *Casamento*

  Nesta estratégia, uma estrutura da resposta esperada à consulta é apresentada e o sistema tenta casá-la com os dados armazenados. Pode ser casamento por exemplo de respostas ou por padrões.

- *Seleção por variação*

  Nesta estratégia, as restrições da consulta são baseadas em variações de valores dos atributos do domínio. Esses valores são mantidos em conjuntos pelo sistema.

Dentre os sistemas que utilizam a navegação como técnica de formulação da consulta podemos citar (AUDDINO *et al.*, 1992) e (CAREY *et al.*, 1996). Utilizando a técnica de subconsultas por agregação de conceitos, podemos citar (TSUDA *et al.*, 1990), que possibilita também a seleção por variação. Como representantes da técnica de casamento de exemplos o mais conhecido é descrito em (ELMASRI e NAVATHE, 1994).

2.1.3 Armazenamento e reuso de consultas

A possibilidade de armazenar consultas é uma funcionalidade indispensável aos sistemas que buscam facilitar o processo de manipulação de dados.

Uma primeira razão para esta importância é a existência de usuários repetitivos, ou seja, que tendem a realizar sempre o mesmo tipo consulta. Se o sistema armazena as consultas, o usuário pode realizar suas consultas a partir de outra já armazenada, alterando apenas seus parâmetros de entrada.

Outra razão importante é a capacidade de se construir consultas complexas a partir de subconsultas. Este processo de construção passo a passo é também muito utilizado por usuários inexperientes, em consultas não muito complexas. Portanto, é uma funcionalidade que auxilia tanto usuários experientes quanto inexperientes no processo de formulação de consultas.

Alguns sistemas permitem essas funcionalidades parcialmente ou completamente, como por exemplo (KUNTZ e MELCHERT, 1989) que permite armazenar consultas que serão visualizadas sob a forma de ícones, podendo ser inseridos em outras consultas; o sistema descrito por AUDDINO *et al.*(1992) permite armazenar consultas para reuso e modificações futuras; outro exemplo muito parecido com o anterior é o sistema descrito por SANTUCCI e SOTTILE(1990), que possibilita guardar as consultas realizadas em uma biblioteca; o sistema descrito por CAREY *et al.*(1996) armazena automaticamente um número específico de consultas realizadas pelo usuário, que podem ser refinadas e executadas posteriormente; outro exemplo é (BEVILAQUA, 1994) que armazena automaticamente a última consulta realizada. Entretanto esses dois últimos sistemas não permitem a construção de consultas novas utilizando as que estão salvas como subconsultas.

2.1.4 Adaptação do sistema ao usuário

Este critério busca avaliar a capacidade do sistema em alterar sua funcionalidade e/ou sua interface de acordo com o usuário. A maioria dos sistemas descritos na literatura não possui este tipo de adaptação, o que existe em alguns sistemas, como (OLIVEIRA, 1992), (KUNTZ e MELCHERT, 1989), (BEVILAQUA, 1994) e (CAREY *et al.*, 1996), neste último apenas uma previsão futura, é a possibilidade de configurar e personalizar o sistema ou ainda disponibilizar funcionalidades extras que auxiliem o usuário.

Todas as facilidades para adaptação dos sistemas ocorrem no sentido usuário sistema, ou seja, o usuário tem a função de executar através do sistema as melhorias disponíveis. O sistema se mantém em uma atitude passiva com relação ao usuário.

Durante os projetos de sistemas de informação, existe uma fase, em que são desenhadas as interfaces do sistema. Durante esta atividade, de forma implícita ou explícita, uma avaliação do público alvo do sistema é realizada. Essa forma de trabalho só funciona quando existe um público alvo fechado, pré-definido. Por

exemplo, para sistemas disponibilizados pela *internet* definir o público alvo é uma tarefa difícil.

CATARCI *et al.*(1991) apresentam um levantamento sobre que aspectos devem ser considerados nos usuários para construção das interfaces dos sistemas de consultas, especificamente. Os aspectos a serem considerados são:

- *freqüência de interação com o sistema*, se são **freqüentes** ou **ocasionais**.
- *variação das consultas*, se são **repetitivos**, ou seja, aqueles que realizam consultas que seguem sempre o mesmo padrão, ou, **extemporâneo**, aquele que requisita informações distintas a cada nova consulta.
- *complexidade estrutural das consultas realizadas*, se as consultas são complexas ou não.
- *conhecimento sobre o domínio de interesse*, mede o quanto do domínio de interesse, esquema a ser consultado, usuário tem conhecimento.

2.2 Sistemas de consultas visuais

2.2.1 QBE

Query-by-Example (QBE) foi a primeira linguagem desenvolvida apoiada no enfoque de facilitar a criação de uma consulta. Nesta linguagem, ao invés da especificação explícita da consulta desejada, o usuário preenche campos nas tabelas com constantes e variáveis, construindo um exemplo da operação desejada. Os resultados obtidos com a execução da consulta são apresentados nas linhas da tabela, onde foi especificada a consulta.

A forma de trabalho neste sistema não é linear, como na forma textual da SQL. Para especificar a consulta o usuário pode movimentar-se por toda a área da tela, caracterizando um estilo bidimensional e além disso o conhecimento que o usuário necessita sobre o domínio da aplicação é minimizado, com relação à SQL, pois toda a relação é apresentada na forma de tabela, entretanto o usuário ainda precisar seguir algumas regras sintáticas para realizar operações que façam sentido.

De acordo com ELMASRI e NAVATHE(1994) o sistema procede da seguinte forma: será apresentado ao usuário uma lista das relações da base de dados para que sejam escolhidas aquelas que serão utilizadas na consulta. A cada relação escolhida, será apresentado um formulário em formato de tabela, com uma coluna contendo o nome da relação e as demais com o nome dos atributos da relação. Nas linhas da tabela, o usuário irá informar o exemplo de operação desejada, utilizando algumas regras sintáticas pré-estabelecidas. As operações possíveis são: *inserção, exclusão, alteração, restrição, junção, agrupamento de valores, agregação de valores e projeção.*

A Figura 2.6 apresenta a consulta realizada no sistema QBE que responde a seguinte pergunta: *Que filmes Allen trabalhou como ator e diretor e estão atualmente passando no cinema Concorde ?*

Filmes	Título	Diretor	Ator
	_X	Allen	
	_Y		Allen

Apresentação	Cinema	Título	Horário
	Concorde	P._X	
	Concorde	P._Y	

Figura 2.6 - Consulta realizada no sistema QBE(ABITEBOUL,1995).

Evoluções da linguagem QBE são utilizadas em diversos sistemas de consultas embutidos em ambientes de desenvolvimento, como os sistemas ACCESS™ e DELPHI™.

Esses sistemas provêem mecanismos gráficos elegantes para construção de consultas conjuntivas. O processo de construção utiliza tanto a representação gráfica formulário, na forma tabular como a linguagem QBE, quanto a representação na forma de diagramas, para apresentar os relacionamentos entre as tabelas que foram selecionadas para a consulta. Essas duas representações gráficas são mantidas isoladas, ou seja, este sistema utiliza uma representação híbrida mas sem construir um novo formalismo.

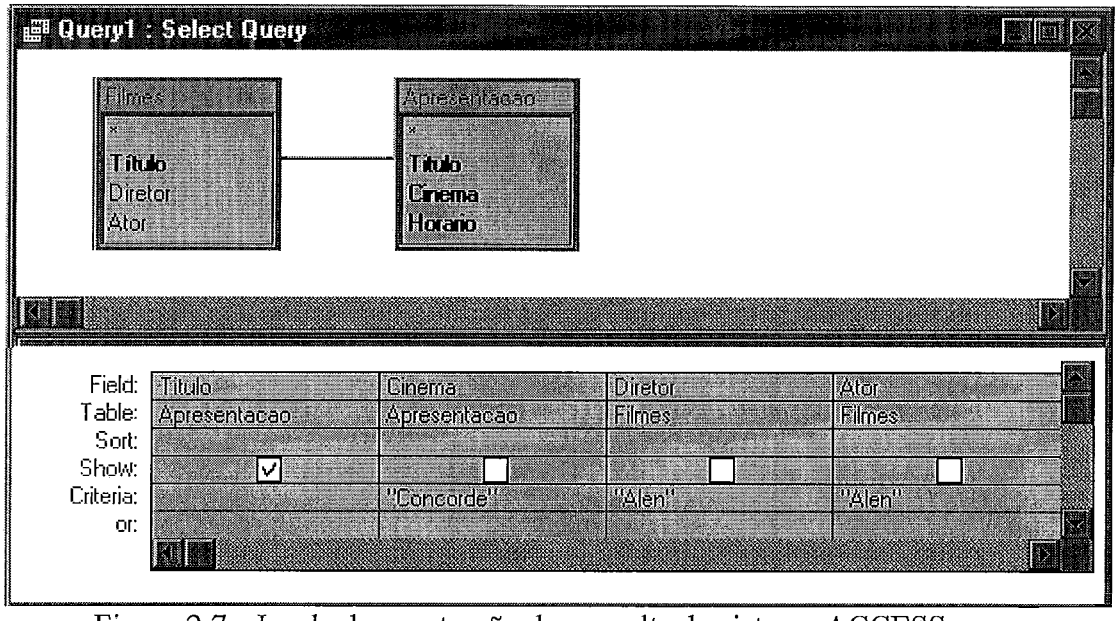


Figura 2.7 - Janela de construção de consulta do sistema ACCESS.

Um exemplo dessa nova forma de utilização da linguagem QBE pode ser visualizada na Figura 2.7 que representa no sistema ACCESS a mesma consulta descrita para a linguagem QBE.

2.2.2 Pasta-3's Graphical Query Language

O sistema construído com esta linguagem, descrita por KUNTZ e MELCHERT(1989), permite a construção de consultas, bem como a navegação pelo esquema do banco de dados. Durante o processo de navegação, o usuário tem acesso a informações sobre as entidades e os relacionamentos, bem como pode visualizar instâncias desses elementos.

Para cada entidade e relacionamento selecionado para a consulta, que podem ser selecionados diretamente ou como resultado do processo de navegação sobre o esquema, é apresentado um formulário de forma tabular com os seus atributos. Selecionando-se o atributo desejado é possível através de opções do menu especificar se o atributo deve constar do resultado, bem como construir condições de restrição da consulta sobre este atributo.

As condições de restrição são apresentadas também de forma gráfica como uma árvore, formada pelos atributos e pelos operadores aplicados. Cada expressão desta árvore constitui um ícone que pode ser manipulado.

Além disso, o sistema permite ainda a utilização dos quantificadores, universal e existencial, bem como a utilização de consultas como subconsulta de outra.

Além de apresentar uma forma gráfica de realização de consultas, esta linguagem possui ainda um outro aspecto bastante interessante que é a tentativa de prover meios de cooperação entre o sistema e o usuário.

Dentre esses meios de cooperação estão a possibilidade de escolher em uma lista construída pelo sistema os valores de constantes a serem utilizados na consulta e ainda de copiar valores apresentados nas janelas do sistema, bem como a construção de consultas sem a especificação completa dos caminhos entre entidades e relacionamentos, pois o sistema é capaz de deduzir, e ainda a capacidade de visualizar ao mesmo tempo resultados de diversas consultas, em janelas distintas, além de permitir a modificação das consultas já realizadas para executá-las novamente.

2.2.3 QBD

Este sistema, descrito por SANTUCCI e SOTTILE (1990), procura facilitar o processo de construção de consultas através da utilização de uma linguagem de consulta visual e de um modelo semântico de dados, onde as relações entre as entidades são explicitamente representadas.

O sistema QBD fornece ao usuário a possibilidade de compreender o domínio de interesse do modelo, através da realização de refinamentos sucessivos sobre o modelo de dados. Desta forma, o usuário pode iniciar seu entendimento do modelo a partir de uma abstração e gradualmente se aprofundar até o modelo real.

Outra característica deste sistema é a criação de subesquemas de interesse, que são na realidade pedaços do modelo da base de dados, que são relevantes ao usuário para a construção de uma consulta.

Os subesquemas podem ser criados através da seleção das entidades no diagrama, a partir de um esquema mantido pelo sistema ou ainda pela localização de conceitos, que pode ser realizada através do estabelecimento de um caminho no grafo, por metacondições sobre os atributos, pela expansão de um subesquema por conceitos vizinhos ou através de um dos refinamentos do esquema.

A construção de uma consulta, neste sistema, é feita a partir da criação de um subesquema de interesse e do estabelecimento do predicado sobre os atributos das entidades representadas neste subesquema.

A Figura 2.8 apresenta o modelo de uma base de dados de vôos. Suponha que se deseja realizar a seguinte consulta: *Quais os pilotos que podem ser parentes de passageiros do vôo X ?*

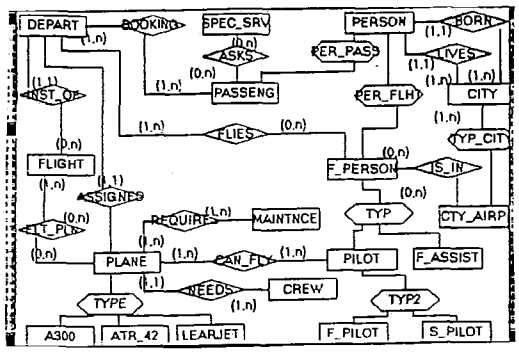


Figura 2.8 - Modelo de dados da base sobre vôos(SANTUCCI e SOTTILE .1990).

A definição da consulta será realizada através da definição do esquema de interesse(Figura 2.9) para esta consulta e a seguir a definição do predicado na janela de restrição(Figura 2.10).

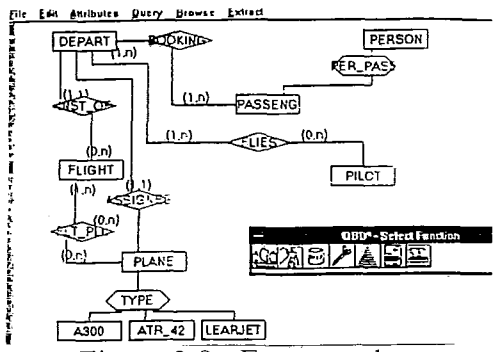


Figura 2.9 - Esquema de interesse(SANTUCCI e SOTTILE ,1990).

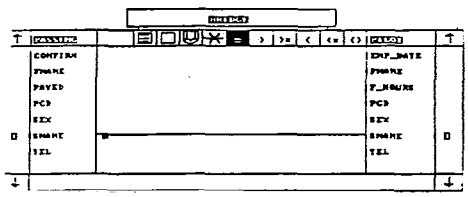


Figura 2.10 - Janela de restrição(SANTUCCI e SOTTILE ,1990).

2.2.4 ICONICBROWSER

Este sistema de consulta visual, descrito por TSUDA *et al.*(1990), utiliza os ícones para representar as entidades a serem consultadas e as operações possíveis. Este sistema visa ser utilizado em banco de dados orientados a objeto e foi desenvolvido também neste paradigma.

A elaboração de uma consulta é feita através da sobreposição de ícones, cuja interpretação é feita de acordo com a seqüência de operações realizadas. A interpretação resulta em uma consulta que pode ser submetida ao banco de dados.

Dentre os principais características do sistema, podemos citar que o usuário não precisa estar familiarizado com o conceito de modelo de dados nem com o

esquema do banco de dados e que as imagens dos ícones sobrepostas possibilita a identificação da consulta.

Este sistema é composto por duas áreas, como mostra a Figura 2.11. A primeira onde se encontram os ícones que representam classes de objetos do banco de dados e a segunda área onde a consulta será construída, através do arrasto de ícones.

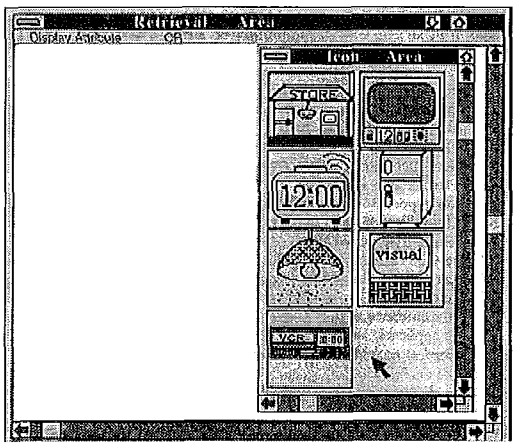


Figura 2.11 - Tela de apresentação do ICONICBROWSER(TSUDA *et al.*, 1990).

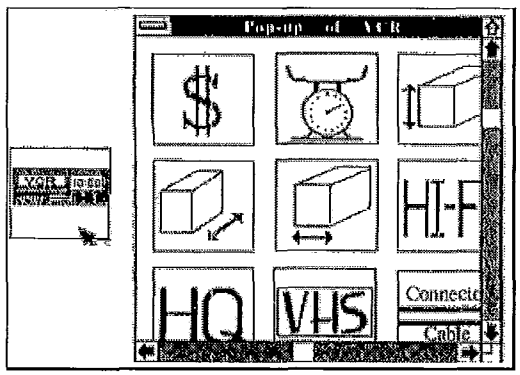


Figura 2.12 - Operação possíveis sobre a classe vídeo cassete(TSUDA *et al.*, 1990).

Os ícones podem assumir três estados distintos: classe, restrição e objeto. No primeiro caso, ele representa um determinada classe de objetos, no segundo caso já foi definido sobre a classe de objetos uma condição, sobre algum ou alguns de seus atributos, e no terceiro caso ele representa objetos do banco de dados.

É possível executar operações sobre os ícones, sendo que algumas são restritas ao estado em que este se encontra. Algumas operações possíveis são:

- *apresentação dos objetos pertinentes ao ícone. Aplicável em objetos em estado classe e restrição.*
- *apresentação das operações disponíveis ao ícone, aplicável em todos os estados.*
- *especificação de união entre ícones em estado classe.*
- *execução de métodos do ícone em estado objeto.*
- *mudança de classe de um ícone em estado classe.*
- *adição de condição ao ícone, para passá-lo do estado classe para o estado restrição.*

A especificação da consulta é feita através do arrasto dos ícones que representam as classes e a aplicação das operações que são apresentadas via *mouse*, como poderá ser acompanhado através do exemplo a seguir.

Suponha um modelo de dados com informações sobre vídeo cassetes, televisões e outros equipamentos, onde o conjunto de operações aplicáveis aos objetos da classe vídeo cassete estão representadas na Figura 2.12. O exemplo descrito a seguir responde a seguinte pergunta : *Qual o peso dos vídeos cassetes com tecnologia Hi-Fi com preço igual ou menor a $600 ?*

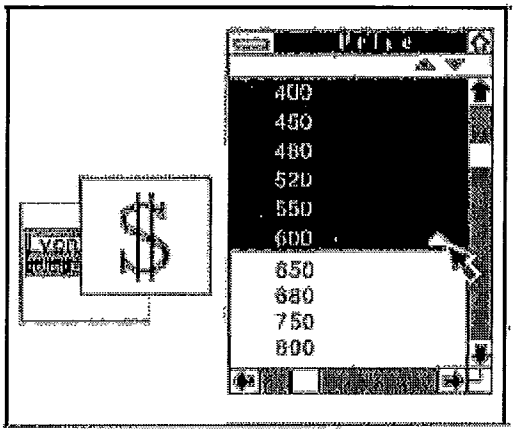


Figura 2.13 - Especificação de parte do predicado da consulta(TSUDA *et al.*, 1990).

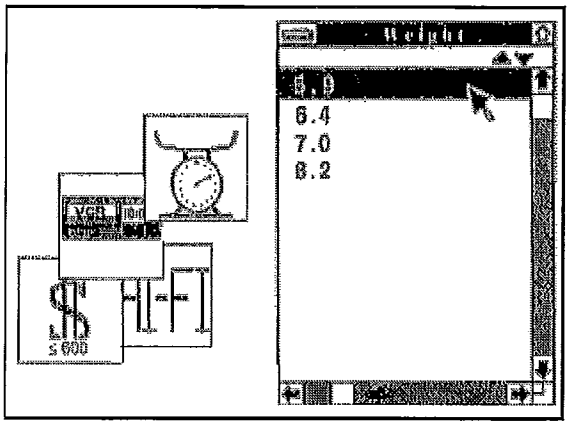


Figura 2.14 - Consulta final no ICONICBROWSER(TSUDA *et al.*, 1990).

Inicialmente, é preciso selecionar o ícone que representa os *vídeos cassetes* na tela inicial do sistema, e aplicar sobre estes o primeiro predicado que será preço menor ou igual a $600. A aplicação deste predicado é feita sobrepondo-se o ícone que representa o *preço* sobre o ícone do vídeo cassete e selecionando-se a variação de valores desejada, como mostra a Figura 2.13.

A seguir, é preciso associar o outro predicado sobre a tecnologia do vídeo cassete, sobrepondo o ícone que representa a tecnologia "*Hi-Fi*" aos dois ícones já sobrepostos, da mesma forma como foi feito anteriormente. Com as restrições da consulta já especificadas é preciso realizar a projeção sobre o atributo *peso*, o que é feito sobrepondo-se o ícone que representa este atributo ao ícone de vídeo cassete. A consulta então será expressa como mostra a Figura 2.14.

Ao selecionar um peso na Figura 2.14, um ícone representando este objeto é criado e sobre este é possível realizar algumas operações, como por exemplo,

visualizar a imagem associada ao ícone. A razão mais importante para a existência deste ícone objeto é sua utilização para realizar associações em outras consultas e desta forma construir consultas de forma incremental.

2.2.5 OdeView

OdeView, descrito por AGRAWAL *et al.*(1990), é a interface gráfica para o banco de dados orientado a objetos Ode. Esta interface possibilita a manipulação do esquema e dos dados e utiliza como representação visual principal os ícones, mas também utiliza os diagramas para apresentar o esquema do banco de dados.

Escolhida uma base de dados para consulta, seu esquema é apresentado em uma janela na forma de diagrama, possibilitando a navegação e o conhecimento do esquema. Selecionando uma entidade é possível visualizá-la de forma mais detalhada, com informações sobre seus atributos, suas superclasses, subclasses e ainda o número de instâncias associadas.

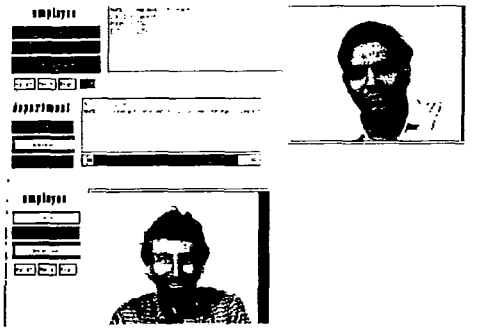


Figura 2.15 - Exemplo de navegação(AGRAWAL *et al.*,1990).

Na janela de informações sobre as classes pode-se iniciar o processo de navegação pelos seus objetos. Através de um botão, é possível apresentar uma janela de conjunto de objetos relativos a classe em questão, onde é possível navegar por este conjunto e pelos objetos a ele relacionados(Figura 2.15), sempre de forma sincronizada. É possível iniciar este processo de visualização de diversas janelas de classes simultaneamente.

De acordo com AGRAWAL *et al.*(1990), somente os processos de navegação pelo esquema e pelos dados estavam implementados no momento da publicação do artigo. Para a realização de consultas que permitissem os processos de seleção e projeção levantou-se o seguinte problema: como determinar sobre quais atributos e métodos das classes seria permitido a realização de projeções e seleções e como estes seriam identificados. A proposta apresentada é a criação de métodos específicos para

a seleção e projeção que informem que propriedades da classe podem ser disponibilizadas para estas operações.

Não foram encontrados na literatura novos trabalhos sobre este sistema. Através de pesquisas na rede *internet* foi possível verificar que o banco de dados *Ode* continua sendo desenvolvido, mas sobre sistema de consulta *OdeView* nada foi encontrado.

2.2.6 SUPER

O sistema SUPER, descrito por AUDDINO *et al.*(1992), consiste em uma ferramenta CASE para definição e manipulação de banco de dados. Suas principais características são a utilização do paradigma da manipulação direta em todos os seus componentes e de um modelo semelhante ao de entidade-relacionamento, estendido ao conceito de objetos, chamado ERC+.

Este ambiente CASE é formado por diversos componentes, e entre este um editor de consultas e uma ferramenta de navegação dos dados. Com o primeiro é possível construir consultas através de interações com o *mouse* enquanto com o segundo componente é possível navegar pelo esquema do banco de dados.

O processo de construção de um consulta no editor de consultas é formado por diversas tarefas, claramente separadas pela ferramenta, através da utilização de janelas específicas para cada uma dessas tarefa. Inicialmente, é preciso especificar a parte do esquema, o subesquema, que será utilizado no processo de consulta, através da cópia dos objetos relevantes da janela de visualização do esquema para a janela de trabalho. A seguir, especifica-se as condições que irão restringir a busca no banco de dados e quais os atributos devem fazer parte do resultado da consulta. Essas duas operações são executadas em uma janela chamada janela de seleção. O resultado obtido com a realização da consulta será apresentado em uma janela, chamada janela de resultados, de forma tabular ou de árvore.

A Figura 2.16 e a Figura 2.17 apresentam um exemplo de consulta, descrito detalhadamente por AUDDINO et al.(1992), que responde a pergunta: *Quais os nomes e endereços das pessoas que possuem seguro do carro Ford ano 1984 ?*

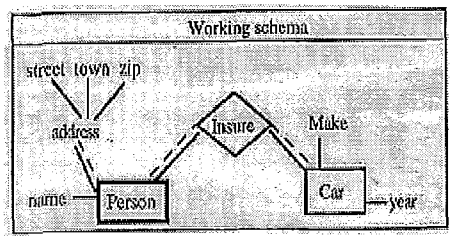


Figura 2.16 - Janela de trabalho do sistema SUPER(AUDDINO *et al.*(1992).

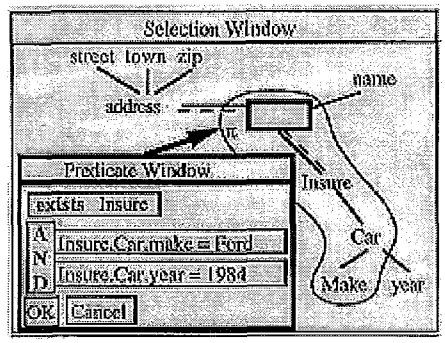


Figura 2.17 - Janela de seleção e de predicado do sistema SUPER.

O sistema oferece também algumas facilidades ao usuário, como a possibilidade de armazenar consultas bem como fazer avaliações parciais, permitindo desta forma evoluir a partir de consultas já realizadas.

2.2.7 GOODIES

OLIVEIRA(1992) descreve o sistema GODDIES, cuja funcionalidade é a apresentação de uma interface gráfica para manipulação do esquema e dos dados para banco de dados orientados a objetos, baseada em múltiplas janelas que utilizam o conceito de manipulação direta.

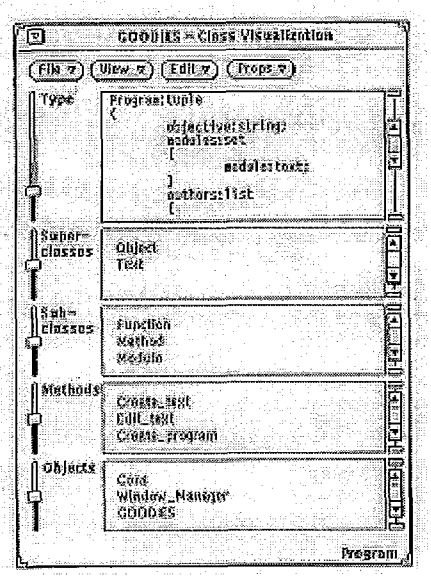


Figura 2.18 - Janela de classe de GODDIES (OLIVEIRA, 1992).

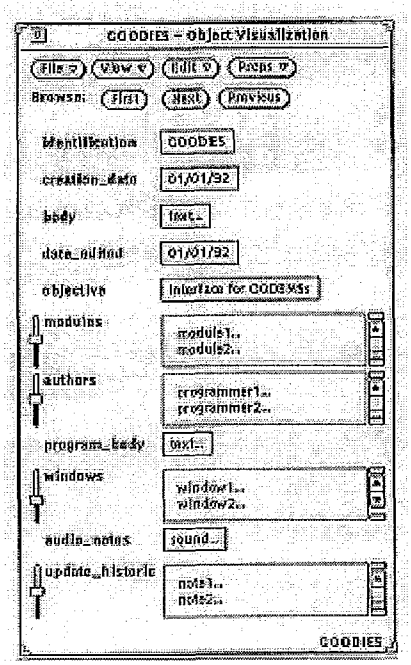


Figura 2.19 - Janela de objetos de GODDIES (OLIVEIRA, 1992).

O sistema possui quatro tipos de janelas básicas que são: a "janela de diretório" que lista as bases de dados disponíveis no sistema de arquivos; a "janela de base de dados" que apresenta as classes definidas em uma base de dados; a "janela de classe" que apresenta a definição das classes através dos seguintes itens: tipo, superclasses, subclasses, métodos e objetos; e a de "janela de objeto" que contém os valores dos atributos de um objeto. Dependendo do tipo destes valores, estes podem ser apresentados na mesma janela do objeto, caso de atributos simples e listas de atributos simples, ou em outra janela, quando o atributo é do tipo texto, imagem, atributo complexo, tuplas e referências para outros objetos.A navegação e a consulta dos dados ocorre a partir da "Janela de classe"(Figura 2.18), selecionando um objeto da opção *"Objects"*, uma "Janela de objeto"(Figura 2.19) será apresentada com os valores dos atributos do objeto selecionado. É possível selecionar diversos objetos e com isso visualizar diversas janelas de objetos ao mesmo tempo.

Na janela de objetos pode-se navegar entre os elementos selecionados e ainda definir o predicado para a consulta, através de uma janela de predicado. As janelas de objetos apresentadas são sincronizadas.

O sistema GODDIES fornece ainda algumas outras facilidades aos usuários que são a possibilidade de se salvar o contexto, ou seja, as janelas de diretório, base de dados e classes; definir níveis de visualização, determinando que atributos e métodos das superclasses e que subclasses deseja visualizar e ainda que informações de classe deseja visualizar na janela de classes.

2.2.8 IRIS

O sistema IRIS é um sistema de banco de dados visual, baseado na apresentação das informações do modelo sobre a forma de gráficos, baseados no modelo ER. Através deste gráfico e de formulários de entrada e saída de dados é que ocorre a interação com os usuários.

Dentre os elementos que compõem este sistema iremos ressaltar AERIAL(Ad-hoc Entity-Relationship Investigation And Learning) descrito por BURNS *et al.*(1993) responsável pelo processo de navegação e GRAQULA descrito por por BURNS *et al.*(1993), responsável pelo processo de edição das consultas.

O subsistema AERIAL é utilizado principalmente por usuários inexperientes no modelo de dados consultado, permitindo que este navegue pelas entidades representadas no grafo, obtendo informações sobre suas instâncias.

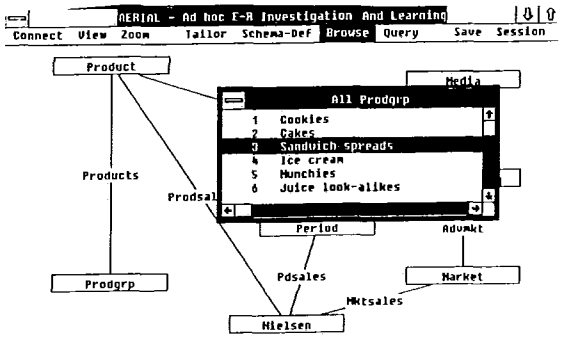


Figura 2.20 - Navegação no componente AERIAL(BURNS *et al.*,1993).

Além disso, é possível selecionar instâncias através de predicados, como pode ser visto na Figura 2.20. A partir das instâncias é possível navegar pelos relacionamentos bem como alterar alguns de seus atributos.

O subsistema responsável pela realização de consultas ao banco de dados foi implementado utilizando-se GRAQULA, que é uma linguagem gráfica para realização de consultas e alterações na base de dados.

Esta linguagem é relacionalmente completa, utiliza técnicas gráficas convencionais como janelas e menus e permite armazenar consultas realizadas para serem reutilizadas e até mesmo se transformarem em uma visão do esquema da base de dados.

A forma de criação e execução de uma consulta é baseada na utilização de diversas janelas que vão determinando as tarefas do processo. Inicialmente o esquema da base de dados é apresentado em uma janela de esquema onde o usuário visualiza as entidades e os relacionamentos que compõem sua base de dados. A seguir, o usuário seleciona as entidades e os relacionamentos que irão compor sua consulta e cria uma ou várias janelas de consulta. Quando a consulta for executada, o resultado será apresentado em uma janela de resultados.

Na janela de consultas, o usuário irá especificar quais os atributos irão ser apresentados no resultado, quais são as condições da consulta e as funções de agregação. A consulta gerada em SQL poderá ser visualizada na janela SQL, permitindo desta forma o aproveitamento desta consulta em outros sistemas.

Não é necessário que o usuário especifique todos os relacionamentos entre as entidades que compõem sua consulta, pois o sistema irá requisitar que o usuário

escolha uma opção de relacionamento antes de executar a consulta, caso estejam faltando conexões. Outra facilidade fornecida pelo sistema é a manutenção do último caminho percorrido na seção de navegação anterior para a próxima seção de navegação. Este caminho pode ser automaticamente transformado em consulta o que caracteriza a reutilização de consultas, mas somente da última realizada.

A especificação da consulta na janela de consulta ocorre de forma muito semelhante à linguagem QBE(ELMASRI e NAVATHE, 1994). Nesta janela, o usuário determinará a apresentação da imagem da entidade ou relacionamento de forma estendida, que consiste em apresentar uma linha para cada atributo, e duas colunas para a colocação de operandos e operadores de comparação.

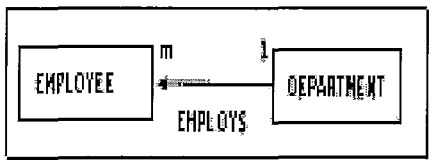


Figura 2.21 - Diagrama da consulta(BURNS *et al.*,1993).

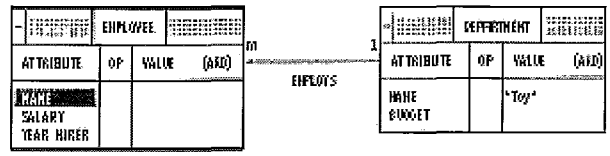


Figura 2.22 - Diagrama expandido com predicado(BURNS *et al.*,1993).

A Figura 2.21 e a Figura 2.22 apresentam um exemplo da linguagem GRAQULA. Este exemplo realiza a seguinte consulta: *Que Empregados trabalham no departamento de Brinquedos ?*

2.2.9 BDE

BEVILAQUA(1994) definiu em sua tese de mestrado um ambiente de Banco de Dados Estatísticos Orientado a Objetos, com utilização bastante intensa de recursos de programação visual, baseado na área de aplicação.

Neste sistema, uma consulta é sempre construída pela definição do **item geográfico**, **corte temporal** e pelas **variáveis de seleção**. Na verdade, podemos compreender essa tríade como o local, tempo e informações. E ainda, é possível após a determinação dessa tríade, aplicar filtros sobre os resultados obtidos.

Para um usuário que deseja consultar o banco de dados estatístico, uma janela inicial é apresentada, onde várias opções são disponibilizadas para construção da consulta. As opções são as seguintes: ***Itens Geográficos*** que permite a determinar os locais geográficos de interesse à consulta; ***Seleção Temporal*** para estabelecer as restrições sobre o atributo tempo dos objetos estatísticos; ***Seleção de Variáveis,*** para

determinar as informações desejadas na consulta; ***Busca de Informações*** que executa a consulta construídas através dos passos anteriores; ***Reprocessar última consulta***, para executar novamente a última consulta; e a opção de ***Auxílio***.

Todas as opções feitas pelo usuário podem ser armazenadas em sua base, podendo ser reutilizadas em outras sessões, mas como nos outros sistemas essa cooperação tem que ser determinada pelo usuário, através da seleção do botão que comanda a gravação das informações.

A Figura 2.23 e a Figura 2.24 mostram a janela de seleção de itens geográficos e a janela de seleção temporal utilizadas durante a definição de consultas neste sistema.

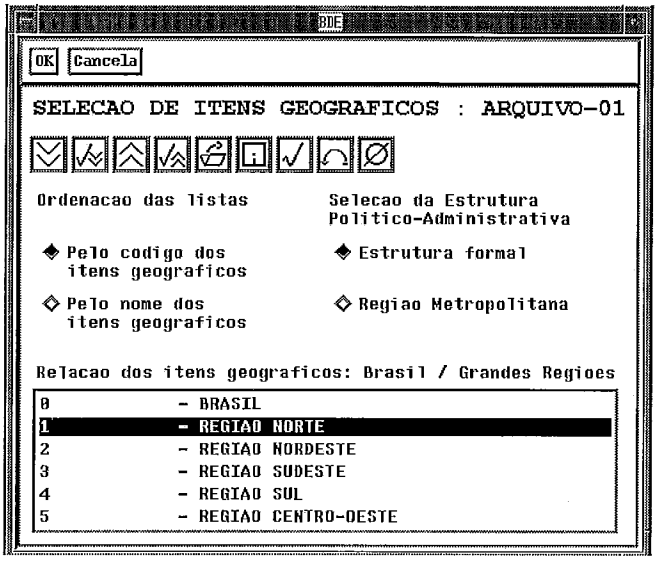


Figura 2.23 - Seleção de itens geográficos (BEVILAQUA, 1994).

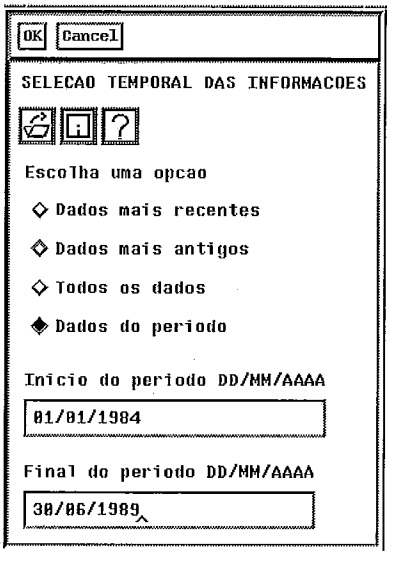


Figura 2.24 - Seleção temporal.(BEVILAQUA, 1994)

2.2.10 SOPView

Este sistema (CHANG *et al.*, 1996) constitui um ambiente gráfico para o banco de dados orientado a objetos SOP.

CHANG *et al.*(1996) defendem que o objeto e suas referências devem ser combinadas e expressas em conjunto, diferenciando-se claramente quando a referência é para um único objeto e quando é para muitos objetos.

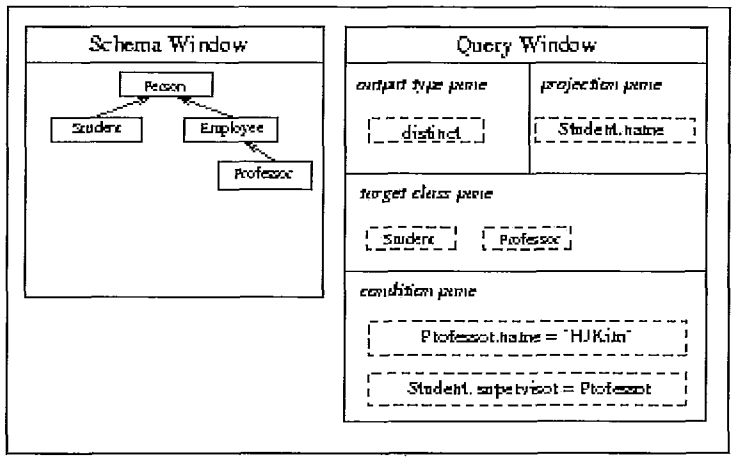


Figura 2.25 - Janelas do sistema SopView (CHANG *et al.*, 1996).

O sistema de consulta aplicado neste ambiente é baseado em uma estrutura de "frames". Duas janelas são apresentadas ao usuário: uma apresentando o esquema a ser consultado e outra para a criação da consulta, como mostrou a Figura 2.25.

A janela de consulta se apresenta como um *frame*, semelhante a um formulário onde os campos devem ser preenchidos com informações para que a consulta possa ser construída. As informações requisitadas são as classes que se deseja consultar, quais os atributos a serem projetados, o predicado da consulta e o tipo do resultado a ser obtido.

Para apresentação do resultado e navegação sobre os objetos existem duas outras janelas. No primeiro caso uma janela, chamada de "Janela de Referências Hierárquicas" é criada para apresentar o resultado da consulta que é feito através de ícones. Este resultado é constituído do objeto buscado e dos objeto(s) relacionado(s). Nesta janela é possível iniciar um processo de navegação.

No segundo caso, para a navegação uma "Janela de Navegação de Objetos" é criada. Neste sistema a navegação também ocorre de forma sincronizada.

2.2.11 PESTO

O sistema PESTO, descrito por CAREY *et al.*(1996), é um sistema gráfico que suporta consulta e navegação de objetos, utilizando representações visuais de formulários e ícones e manipulação direta dos objetos através da interface. Este sistema não é relacionalmente completo.

Este sistema é o primeiro na literatura implementado para o modelo orientado a objetos que se diz portável, pois gera a consulta em linguagem OQL, padrão ODMG

(CATTEL e BARRY,1997). Além disso é extensível, permitindo a inclusão de novos predicados e formas de apresentação dos objetos.

Selecionada a coleção a ser utilizada na consulta, uma janela de navegação com o primeiro objeto da coleção é apresentada. Esta janela possuí barra de opções que permite a navegação pelos objetos desta coleção, a realização de consultas, a visualização do objeto como um membro de sua classe mais específica e a visualização simultânea de vários objetos da coleção na forma de tabela.

A opção de consulta da barra de ferramentas permite a especificação do predicado da consulta. Ao se escolher esta opção, a janela passa para o modo de consulta onde a barra de opções é alterada e os atributos de objeto estão aptos a receber predicados. Definidos os predicados, deve-se pressionar o botão da barra de opções que dispara a consulta e a janela voltará ao modo de navegação, onde somente os objetos da coleção que atenderem ao predicado estarão disponíveis para visualização.

Os objetos associados aos objetos apresentados podem ser visualizados através da seleção dos botões que representam seus relacionamentos. O sistema implementa a sincronização das consultas e também armazena um histórico das consultas já realizadas para cada janela e as torna disponíveis quando a janela está em modo de consulta.

A Figura 2.26 apresenta a consulta *Quais os estudantes GPA > 3,5 que estão assistindo a um curso de CS ministrado por um professor da área de Banco de Dados ?* realizada neste sistema.

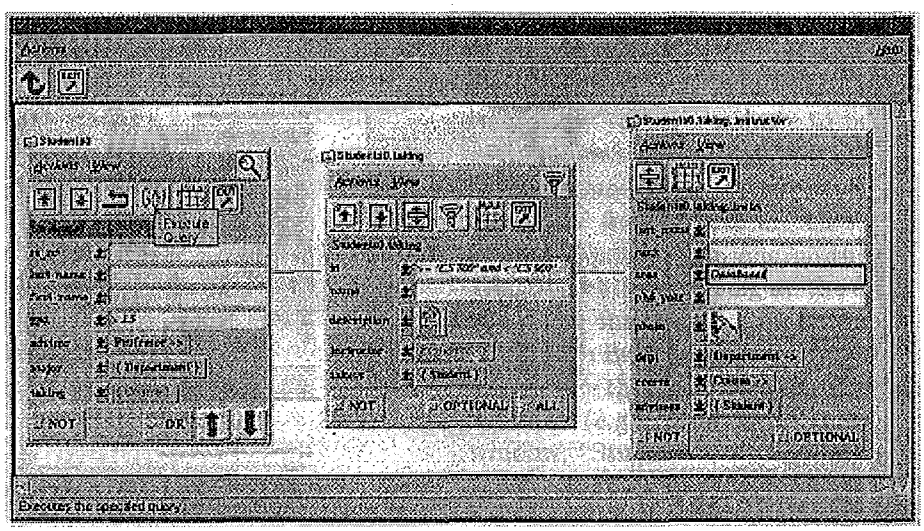


Figura 2.26 - Janela de consulta(CAREY *et al.*,1996).

Uma característica bastante interessante deste sistema é a possibilidade da realização de junções por valor e por referência. O processo foge um pouco do padrão descrito anteriormente, pois uma janela sincronizadora será criada para controlar a navegação pelos objetos selecionados de acordo com a junção criada.

Outra característica única deste sistema é a possibilidade de especificar métodos nos predicados de tipos especiais de dados, como imagem e texto.

2.3 Conclusão

Este capítulo apresentou o último estágio de evolução no processo de realização de consultas na perspectiva do usuário. ZLOOF(1996) apresentou uma evolução das ferramentas visuais para banco de dados e programas de aplicação sob a perspectiva dos desenvolvedores, ressaltando as ferramentas e linguagens de programação desenvolvidas desde os anos 60 utilizadas para o desenvolvimento de aplicações de banco de dados, destacando sempre as evoluções obtidas nas interfaces desenvolvidas.

Neste capítulo, apresentamos o conceito de sistemas de consultas visuais, e algumas características que podem ser utilizadas para classificar estes sistemas como a representação visual utilizada, tanto para a formulação da consulta quanto para apresentação do resultado, a forma de conhecer o domínio de interesse e de construir as consultas. Além destes que classificam os sistemas, existem outros fatores que devem ser verificados nestes sistemas que são: a possibilidade de reutilizar consultas e se o sistema fornece algum tipo de adaptação ao usuário.

A maioria dos sistemas desenvolvidos facilita a realização de consultas apenas no nível de interface, não se preocupando em fornecer funcionalidades extras para facilitar o processo.

Os sistemas baseados na representação visual formulários são bastante simples de serem utilizados, mas dificultam a formulação de consultas sofisticadas, bem como reduzem a produtividade de usuários com conhecimento sobre a área de banco de dados.

Já os sistemas baseados em diagramas são difíceis de serem utilizados por usuários inexperientes que não conhecem o domínio de interesse. Entretanto são perfeitos para usuários com conhecimento, pois permitem a elaboração de consultas sofisticadas.

Os sistemas que utilizam os ícones são mais adequados para os usuários inexperientes que realizam consultas simples. A dificuldade na utilização desta representação visual está na associação da imagem e o conceito desejado.

A criação de um módulo para que o usuário aprenda o domínio de interesse só faz sentido para sistemas destinados a usuário inexperientes.

É possível obter um sistema que possa ser utilizado por grupos heterogêneos de usuários através de algum tipo de adaptação do sistema de acordo com características do usuário, possibilitando escolher o que apresentar e como apresentar, de acordo com os conhecimentos do usuário. Entretanto o máximo que se fez até hoje e pode ser recuperado na literatura neste aspecto de adaptação ao usuário são configurações e personalizações das interfaces do sistema, entretanto este processo é disparado pelo próprio usuário (OLIVEIRA, 1992, CAREY *et al.*, 1996).

Para finalizar este capítulo, vamos apresentar em seguida duas tabelas comparativas dos sistemas de consultas visuais avaliados. A Tabela 2.1 apresenta as operações da linguagem de consulta implementadas pelos sistemas. A Tabela 2.2 apresenta a classificação dos sistemas de acordo com os critérios descritos neste capítulo.

Sistemas	Modelo de Dados	Operações Implementadas ou Definidas para Implementação
QBE	Relacional	relacionalmente completo.
Access	Relacional	relacionalmente completo.
Pasta'3	Relacional	seleção, projeção, junção, agregações, quantificadores.
QBD	Relacional	relacionalmente completo.
IconicBrowser	OO	seleção, projeção, junção, união, consultas aninhadas, agregações, aplicação de mensagem aos objetos.
OdeView	OO	seleção, projeção, junção.
SUPER	Relacional & OO	criação e alteração de classes e objetos; seleção, projeção, quantificador existencial.
GOODIES	OO	seleção, junção.
IRIS	ER & Relacional	relacionalmente completo.
BDE	OO	seleção, projeção.
SOPView	OO	seleção, projeção, junção, consultas aninhadas.
PESTO	OO	seleção, junção, disjunção, quantificadores.

Tabela 2.1 – Tabela das operações de consulta dos sistemas descritos.

Sistemas	Modelo BD	Representação Visual	Compreensão do domínio	Formulação de consulta	Reuso de Consultas	Adaptação para o usuário
QBE	Relacional	Formulários	Não possuí	Casamento	não	não
Access	Relacional	Diagramas & Formulários	Não possuí	Casamento	não	não
Pasta'3	Relacional	Formulários & Diagramas	Não possuí	Casamento Subconsultas	sim	Funções de auxílio
QBD	Relacional	Diagramas	Refinamento e seleção	Navegação & Subconsultas	sim	não
IconicBrowser	OO	Ícones	Navegação	Subconsultas	não	não
OdeView	OO	Diagramas & Ícones	Navegação	Navegação	não	não
SUPER	Relacional & OO	Formulários & Diagramas	Navegação	Navegação	sim	não
GOODIES	OO	Formulários	Navegação	Navegação	não	não
IRIS	ER & Relacional	Formulários & Diagramas	Navegação	Navegação	não	não
BDE	OO	Formulários & Ícones	Não possuí	Subconsultas	sim	Funcionalidades para auxílio
SOPView	OO	Formulários, Diagramas & Ícones	Não possuí	Navegação Subconsultas	não	não
PESTO	OO	Formulários, Diagramas & Ícones	Não possuí	Navegação	sim	não

Tabela 2.2 – Tabela comparativa dos sistemas de consultas visuais.

Capítulo 3 - Interfaces Adaptativas

As interfaces adaptativas são caracterizadas por realizar alterações em sua forma e/ou conteúdo de acordo com o conhecimento adquirido sobre o usuário que utiliza a interface.

A capacidade da interface em tomar decisões relativas à forma de comunicação com o usuário visa torná-la mais fácil de ser utilizada, melhorando o desempenho do usuário na realização de suas tarefas.

De acordo com as notas de um seminário de WARN(1997), a adaptação da interface é uma técnica utilizada para criar interfaces inteligentes como também a modelagem do usuário, a linguagem natural, modelagem de diálogos e o gerador de explicações.

Um forma bastante comum de implementação das interfaces adaptativas é através da tecnologia de agentes inteligentes. Na literatura costuma-se considerar as interfaces adaptativas como um tipo de agente inteligente, como os tutores inteligentes e os assistentes pessoais. Entretanto, consideramos que existem formas de implementar esse tipo de interface sem utilizar os conceitos inerentes aos agentes inteligentes e por isso descrevemos essas tecnologias separadamente neste capítulo.

Este capítulo encontra-se dividido em 4 sessões. A primeira sessão realiza uma introdução ao conceito e ás características de interfaces adaptativas. A sessão seguinte trata dos agentes inteligentes como uma forma de implementação das interfaces adaptativas. Na terceira sessão apresentamos formas de se modelar o usuário indispensáveis para a construção de interfaces adaptativas e finalmente na última sessão apresentamos algumas conclusões sobre este capítulo.

3.1 Introdução

As interfaces adaptativas são caracterizadas por sua capacidade de modificação em resposta a características ou necessidades do usuário, identificadas pelo sistema. Constituem uma nova forma de interação entre o homem e a máquina.

De acordo com HÖÖK(1997), essa interface foi proposta na tentativa de solucionar problemas que o paradigma de manipulação direta não foi capaz, como por exemplo: problemas com grande volume de informações, fornecimento de auxílio no uso de sistemas complexos e problemas de tempo real.

A adaptação busca tornar a interface mais amigável ao usuário, facilitando sua utilização e melhorando o desempenho do usuário ao realizar suas tarefas. Outras vantagens obtidas neste tipo de interface inteligente são a redução do tempo de aprendizado e do número de erros cometidos pelo usuário.

De acordo com DOWNTOM(1992) a adaptação do sistema está baseada nas diferentes características dos usuários e de seus comportamentos que se modificam com o tempo. Esse tipo de sistema está intimamente ligado a modelagem e categorização de seus usuários.

Entretanto, a criação e utilização de interfaces adaptativas apresenta algumas desvantagens que devem ser consideradas durante o processo de elaboração e implementação:

- *dificuldade de desenvolvimento*
- *falta de previsão do sistema*
- *falta de confiança do usuário no sistema em virtude de suas mudanças*
- *interfaces dinâmicas com alterações dependentes de conhecimento*
- *mapeamento do conhecimento real e dinâmico de forma discreta*

O livro "Adaptative User Interface"(BROWNE *et al.*, 1990) descreve três formas distintas de realizar a adaptação da interface:

- *a interface coleta informações sobre o usuário e se altera na mesma sessão ou entre sessões.*
- *a interface identifica o usuário como pertencente a uma categoria e determina as alterações de acordo com a categoria identificada.*

- *a interface não é alterada, mas ocorre uma alteração no desempenho, através de um tratamento de erros mais eficiente.*

A fim de construir uma interface adaptativa, independente da escolha de uma das formas citadas acima, é necessário manter uma modelagem do usuário acompanhada ou não de histórico de utilização do sistema.

As interfaces que se adaptam a cada usuário de forma individual devem manter modelos individuais, enquanto as interfaces que reconhecem o usuário como pertencendo a uma categoria, devem manter modelo para cada categoria a ser tratada pelo sistema e possível de ser identificada no público alvo.

Em sistemas onde a interface não se altera, mas o desempenho evolui deve-se manter um modelo do usuário que armazene seus erros mais comuns e atitudes a serem tomadas para que o desempenho possa ser melhorado.

STEPHANIDIS *et al.*(1993) citam algumas dimensões a serem consideradas para a construção de um sistema com interface adaptativa: o nível e o momento de se realizar a adaptação, qual a parte do sistema que irá controlar esta adaptação e o tipo de conhecimento que é necessário para realizar as adaptações entre outros. Esses fatores surgem em virtude da associação do conhecimento ao processo de elaboração de interfaces.

A adaptação da interface de acordo com o usuário pode ocorrer em dois sentidos: da interface para o usuário e do usuário para a interface. No primeiro caso a interface reconhece o usuário e/ou sua categoria e se altera, para se tornar melhor. Em alguns sistemas, a adaptação só é efetivada após a aprovação do usuário. No segundo caso, o usuário seleciona entre opções possíveis de visualização da interface aquela que melhor se adapta a ele. Outra possibilidade neste sentido é a capacidade do sistema perceber as falhas do usuário e sugerir treinamentos, de forma a adaptar o usuário à interface.

Como exemplo de interfaces adaptativas desenvolvidas e retratadas na literatura podemos apresentar a barra de ferramentas adaptativa (MIAH *et al.*, 1997), o AVANTI (FINK *et al.*, 1996), DiBlue (DATTOLO e LOIA, 1997), PUSH(ESPINOZA e HÔÔK, 1996) e o Modelo de Relevância(MATHE e CHEN, 1994).

Barra de ferramentas adaptativa é um sistema desenvolvido por MIAH *et al.*(1997) que opera sobre o *Microsoft Office* e provê a inserção de barras de ferramenta de acordo com as necessidades do usuário capturadas pelo sistema. A inteligência desta

interface encontra-se não só em reconhecer a necessidade de inserção de uma nova barra, mas também em decidir pela remoção de uma barra de ferramenta da tela, quando o número de barras apresentadas ultrapassa o limite de três. Esta remoção é feita de acordo como o grau de importância das barras apresentadas para o usuário, sendo aquela de menor importância retirada da tela. A determinação do grau de importância é baseada em informações armazenadas na base de fatos relativas ao momento de criação da barra, à última interação e à freqüência de sua utilização.

Outro exemplo de sistema com interface adaptativa é o sistema AVANTI apresentado por FINK *et al.*(1996). Este sistema realiza a adaptação do conteúdo e da apresentação de páginas *web,* de acordo com as características do grupo que o usuário pertence e de suas próprias características. Este projeto visa prover informações hipermídia contidas em computadores de uma área metropolitana, como os da polícia, hospitais, transporte, entre outros, para diversos tipos de usuários: turistas, moradores, agentes de viagens e outros.

O DiBlue apresentado em DATTOLO e LOIA(1997) é um ambiente hipermídia desenvolvido de acordo com uma arquitetura baseada no modelo de atores, do tipo agentes, que pretende desenvolver interfaces hypermídia adaptativas em sistemas cujos serviços e dados estão distribuídos. Esta arquitetura é baseada na existência de quatro agentes: *StorActor, TeleoActor, InfoActor e UserActor.* Esses agentes são responsáveis pelo processo de monitoramento do comportamento do usuário, determinação de novas preferências e conseqüente adaptação da interface. O agente *StorActor* contém um objeto hipermídia e os serviços correspondentes. O agente *TeleoActor* realiza a comunicação entre o objeto hipermídia e o usuário. Os dois últimos agentes são responsáveis pela adaptação, pois monitoram o comportamento do usuário e determinam as possíveis adaptações.

O sistema PUSH descrito por ESPINOZA e HÖÖK(1996) propõe uma forma inteligente de apresentação de grandes volumes de informações. Para isso, utiliza um domínio específico, a documentação do método de desenvolvimento de software OO, chamado SDP, composta por mais de 500 documentos com número de páginas variando de 5 a 20. A interface deste sistema é baseada em páginas html geradas pelo sistema com formas de interação multi-modo, pois contém textos estáticos, textos manipuláveis pelo usuário e grafos que também podem ser manipulados pelo usuário e utilizados para navegar pelas informações textuais, além de permitir a realização de perguntas.

A adaptação neste sistema pode ocorrer em duas direções: o usuário pode modificar o texto apresentado através da remoção e expansão dos tópicos apresentados e realizar perguntas; o sistema apresenta informações que considera importantes para o usuário baseado na tarefa realizada. Esta tarefa pode ser informada pelo usuário ou inferida pelo sistema baseada em regras e em uma hierarquia de tarefas obtida da observação do comportamento dos usuários em situações reais e são do tipo planejamento de projeto, aprendizado de método, execução de atividades, produção de produtos e outras.

O Modelo de Relevância descrito por MATHE e CHEN(1994) constitui um modelo de aprendizado da relevância contextual do usuário durante a recuperação de informações. Este modelo pode ser utilizado em conjunto com os mecanismos de busca disponibilizados na rede.

Neste modelo o usuário tem controle sobre todo o processo de adaptação facilitando seu entendimento e aceitação. O controle é feito através da informação ao sistema se o que foi recuperado é relevante ou não para o usuário no contexto corrente, que foi selecionado pelo usuário ao iniciar a consulta. A informação ao sistema de sua satisfação com as informações recuperadas é feita de forma bem simples para não desviar o usuário de sua real tarefa.

As redes de relevância são construídas a partir dessas informações de satisfação do usuário conectando os pontos de entrada(consultas) com os pontos de saída(páginas recuperadas) e associando às conexões valores de relevância, para que em consultas futuras o sistema possa decidir a partir desta rede se uma página selecionada deve ou não ser apresentada ao usuário.

3.2 Agentes Inteligentes

O conceito de agente inteligente ainda não é muito padronizado na literatura. Segundo WOOLDRIDGE e JENNINGS(1995), existem dois conceitos formados: um primeiro chamado fraco e o segundo forte.

O primeiro conceito de agentes inteligentes considera que o termo denota um *hardware* ou *software* que apresenta as seguintes propriedades:

- ***autonomia***, ou seja, capacidade de atuar sem a manipulação direta de um usuário.

- ***habilidade social,*** ou seja, possibilidade de interação com outros agentes.
- ***reação,*** ou seja, capacidade de perceber o que ocorre no ambiente e reagir em um tempo razoável.
- ***iniciativa,*** ou seja, capacidade de apresentar um comportamento orientado ao seu objetivo, não agindo somente em resposta ao ambiente.

Segundo ETZIONI e WELD(1995), além dessas propriedades descritas anteriormente os agentes devem possuir:

- ***continuidade temporal,*** ou seja, ser um processo que roda continuamente.
- ***adaptabilidade***, ou seja, deve ser capaz de se modificar de acordo com as preferências do usuário bem como de se adaptar ao ambiente em que se encontra.
- ***mobilidade,*** ou seja, deve ser capaz de se movimentar entre máquinas de diferentes arquiteturas e plataformas.

É praticamente impossível, no estágio de desenvolvimento atual, obter-se um agente que possua todas as propriedades citadas. A combinação de algumas delas já é suficiente para construção de um agente inteligente.

Normalmente no desenvolvimento de sistemas inteligentes utiliza-se uma base de conhecimento, ou seja, uma base com informações sobre o mini-mundo que está sendo modelado e refletido pelo agente. A obtenção do conhecimento e a geração das verdades absolutas do mundo do agente, conhecidas como fatos, são as duas tarefas consideradas mais difíceis pelos desenvolvedores.

Segundo ZWICKER e REINHARD(1990), a base de conhecimentos de um agente inteligente pode possuir categorias de conhecimento de três tipos distintos, de acordo com a funcionalidade do agente.

As categorias de conhecimento dos agentes inteligentes são:

- *habilidades genéricas*

  Inclui informações capazes de orientar o agente em suas tarefas. Como exemplo podemos citar estilo de ensino, estratégias de recomendações e habilidades com linguagem natural.

- *conhecimento do assunto*

  Este tipo de conhecimento se refere ao assunto tratado pelo agente, como procedimentos que podem ser tomados, metas a serem atingidas e recursos disponíveis.

- *modelos de usuário*

  Neste tipo de conhecimento se enquadram informações obtidas através da utilização do sistema pelo usuário, que demonstrem suas preferências e necessidades.

  Segundo MAES(1994), é possível prover ao agente a capacidade de adquirir esse conhecimento através do aprendizado do comportamento do usuário. Para que isso seja possível, é necessário que o uso da aplicação envolva um comportamento repetitivo, por parte das ações efetuadas pelo usuário, bem como, que este comportamento seja distinto entre os usuários.

Além da base de conhecimento é necessário manter no agente inteligente uma máquina de inferência, ou seja, um componente de sistema capaz de realizar perguntas a base de conhecimento e obter respostas. Uma forma muito comum de implementar essa máquina é utilizando regras que são formas de expressar o conhecimento através de um conjunto de sentenças relacionadas.

3.2.1 Considerações

De acordo com NORMAN(1994), para se garantir a introdução desta nova tecnologia(agentes), é necessário considerar dois aspectos importantes: primeiro, como as pessoas se sentem sobre os agentes e segundo, como são aceitas as ações automáticas, autônomas, executadas pelos agentes.

Alguns fatores devem ser considerados, como:

- *manter o sentimento de controle das pessoas*

  Manter o controle das atividades realizadas é um aspecto psicológico bastante importante, que deve ser considerado ao se implementar e implantar um agente inteligente. As pessoas devem se sentir confortáveis com as ações efetuadas pelo agente. Uma forma de se garantir essa confiança é manter um modelo das ações efetuadas

pelo agente, permitindo aos usuários compreender, acompanhar e até desfazer essas ações.

- *não gerar expectativas fora da realidade*

  Existe uma tendência nas pessoas em criar expectativas exageradas sobre as capacidades de um agente inteligente. Este problema se torna ainda maior quando formas humanas são utilizadas nos agentes, gerando com isso a esperança de comportamentos, pensamentos e ações humanas.

- *garantir a segurança e a privacidade*

  Os conceitos de segurança e privacidade estão embutidos no sentimento de controle que as pessoas necessitam manter. Entretanto, esses aspectos são difíceis de serem garantidos, pois como garantir que os agentes não irão agir provocando problemas físicos, mentais ou monetários aos usuários, uma vez que eles podem entrar no sistema vindo de outras máquinas, via e-mail, por exemplo.

  Os agentes possuem acesso a registros pessoais, correspondência, além de executarem atividades financeiras, impossibilitando a garantia da privacidade.

- *a interação entre o homem e o agente*

  Ainda de acordo com NORMAN(1994), ainda não foi definida uma forma de interação entre os agentes e as pessoas que seja considerada satisfatória, seja feita de forma implícita pelo agente ou através de pequenas linguagens de programação declarativas ou gráficas.

  O termo interação diz respeito à forma como os agentes serão instruídos e controlados, como será o retorno do agente para o usuário, como o agente irá oferecer conselhos e informações e ainda como o modelo conceitual do usuário dos métodos dos agentes para as operações e atividades será apresentado.

3.3 Modelos de Usuários

Os modelos de usuários são utilizados pelas interfaces adaptativas a fim de possibilitar a manutenção de informações sobre o usuário. Essas informações podem

ser simples, como um simples número indicando o grau de conhecimento de determinado conceito, como estruturas complexas, representando o conhecimento de um assunto. Normalmente, o tipo de informação armazenada corresponde a um histórico da utilização do sistema pelo usuário e de algumas características pessoais, que possam ser úteis na tomada de decisão pelo sistema

Este conceito surgiu para atender as necessidades dos tutores inteligentes, programas de computador capazes de individualizar suas ações, a partir de inferências sobre o conhecimento do estudante, com a finalidade de ensiná-lo ou treiná-lo (MACIEJEWSKI e KANG,1994). Os tutores precisam manter um acompanhamento da evolução do aprendizado dos alunos, no uso do sistema, tornando-se conhecidos como modelo de alunos. Com o modelo, o tutor é capaz de ensinar ao aluno baseado em conceitos que ele já mostrou saber, bem como, supor que o aluno sabe determinado conceito, a partir de seu conhecimento sobre outros conceitos.

Alguns sistemas tutores, como o descrito por SELKER(1994), categorizam seus estudantes de acordo com o nível de conhecimento demonstrado sobre o assunto a ser tutorado. Normalmente, esta categorização é feita em quatro níveis: novatos, intermediários, profissionais e especialistas. De acordo com a categoria em que o estudante se encontra, formas distintas de auxílio são fornecidas.

Mesmo antes do surgimento dos sistemas pessoais, ou seja, sistemas que reagem em função do usuário, os desenvolvedores de sistemas já desenvolviam modelos de seus usuários. Entretanto, este modelo era mental e normalmente não individualizado, destinado ao grupo de usuários do sistema.

Durante a utilização de um sistema de informação baseado em diálogos, existem diversas formas de se obter informações sobre o usuário que abasteçam o modelo. KOBSA e WAHLSTER(1989) ressaltam as seguintes:

- *fatos padrões determinados pelo sistema*
- *modelos iniciais do usuário mantidos pelo sistema*
- *fatos obtidos a partir da interação do usuário com o sistema*
- *fatos obtidos a partir da resposta do usuário ao sistema*
- *fatos obtidos a partir do usuário mas através de mecanismos diferentes do lingüístico.*

MAES(1994) descreve ainda outra possibilidade de aquisição de conhecimento:

- *troca de conhecimento, agentes que realizam a mesma tarefa podem trocar ou compartilhar conhecimentos.*

ZISSOS e WITTEN(1985) e SLEEMAN(1985) apresentam em seus trabalhos dimensões importantes que devem ser considerados durante o processo de elaboração de um modelo de usuários:

- *número de usuários aos quais o modelo se aplica*

  O modelo pode representar o conhecimento sobre um único usuário ou ser construído de forma canônica, podendo ser aplicado a uma categoria de usuários.

- *o momento de formação do modelo*

  O modelo pode ser construído a partir de informações fornecidas pelo usuário para descrevê-lo ou para descrever suas necessidades ou então através da observação pelo sistema das interações do usuário.

- *abrangência do modelo*

  O modelo pode ser formado por conhecimentos com grande tempo de vida ou com informações específicas para determinada tarefa.

- *informação contida no modelo e como tratá-las*

  As informações de conhecimento do usuário podem ser modeladas de diversas formas, como por exemplo, por números representando a complexidade de uma tarefa; através de pesos; como sub-conjunto do conhecimento de um especialista; e ainda como resultados da tarefa produzidos pelo sistema para serem comparados aos resultados do usuário. Cada uma dessas formas de informação é tratada de forma diferente pelo sistema.

A forma de construção do modelo de usuário está intimamente relacionada à tarefa que será desempenhada e por isso não existe um modelo melhor do que outro. Muitas vezes, é preciso uma combinação de tipos de modelos para atender plenamente as necessidades de um sistema.

DOWNTOM(1992) apresenta uma classificação para os modelos, baseada na forma como são mantidos e utilizados.

- *Modelos estáticos alteráveis*

  O modelo do usuário não é alterado e um mesmo modelo inicial é utilizado para todos os usuários. As diferenças entre os usuários são modeladas a partir da manutenção de um histórico de utilização do sistema e é sobre este histórico que são determinadas as diferenças entre os usuários. Este histórico é mantido no modelo.

- *Modelos de comparação*

  São utilizados dois modelos, sendo um estático não alterável e outro que pode ser alterado. O modelo estático reflete o comportamento de um usuário especialista e é utilizado para comparar com o modelo do usuário para a tomada de decisões.

- *Modelos Alternativos*

  São modelos estáticos que caracterizam um tipo de comportamento. A partir de um conjunto de regras, um modelo deste tipo é selecionado para ser utilizado para um usuário. Este tipo de modelagem considera que as características do usuário não são freqüentemente alteradas.

- *Modelos baseados no reconhecimento de planos*

  Este tipo de modelo, na realidade, não modela o usuário e sim as tarefas que ele irá desempenhar. É comum sua utilização em interfaces adaptativas.

- *Modelos de utilização*

  Neste modelo, as diferenças na utilização da informação no sistema são consideradas, baseadas na quantidade de utilização de determinado item. As diferenças entre os usuários não são consideradas.

LUCENA e GUARANYS(1995) propõem um novo tipo de modelo de usuário, baseado na utilização de dois modelos estáticos, chamados de estereótipos, como apresentado a seguir:

- *Modelo de dois estereótipos*

  O primeiro modelo estereotipado representa características comuns de um grupo de usuários e o segundo representa o

conhecimento sobre a tarefa a ser realizada. Baseado nesses dois modelos e na utilização do sistema pelo usuário é possível construir o modelo do usuário.

3.4 Conclusão

Na literatura, podemos encontrar algumas propostas de interfaces adaptativas, principalmente associadas a *internet* e a sistemas hipermídia, inclusive de ambientes de desenvolvimento de sistemas capazes de construir de forma automática interfaces adaptativas, como o ambiente GAIA descrito em (MARTINS, 1996) e o TRIDENT descrito em (VANDERDONCKT e BODART, 1993).

A escolha do melhor modelo de usuário para um determinado sistema está vinculada às tarefas a serem desempenhadas por este e no nível de confiabilidade desejada. Assim, por exemplo, se desejamos que um sistema tenha um nível de confiabilidade alto, no que diz respeito à satisfação do usuário com as atitudes do sistema, não devemos escolher modelos estáticos baseados simplesmente em categorias de usuário.

É importante manter sempre em mente que os usuários tendem a evoluir durante a utilização de um sistema, à medida que seu aprendizado cresce. Desta forma, consideramos indispensável a manutenção de um histórico de utilização do sistema no modelo do usuário, para que essas mudanças de comportamento possam ser refletidas no sistema.

Capítulo 4 - Arquitetura para Sistemas de Consultas Visuais em SGBD-OO$_S$ com Interfaces Inteligentes

Os sistemas de banco de dados têm evoluído bastante nas últimas décadas, no que diz respeito ao modelo de dados utilizado, bem como ao seu público consumidor. Esta variedade de usuários torna difícil a criação de interfaces e escolha de paradigmas de interação capazes de atender plenamente a todos os possíveis usuários do sistema.

Atualmente, com o consumo de micro computadores como bem doméstico e com a grande interação possibilitada pela rede mundial de computadores (*internet*), o número e a variedade de usuários cresceu bastante. Os usuários consumidores de informações de bases de dados apresentam diversas categorias de conhecimento, e podem variar desde adolescentes em busca de informações para pesquisas escolares, pesquisadores das mais diversas áreas de interesse, médicos, jornalistas, advogados e entre outros. Esses profissionais não carregam consigo conhecimentos sobre a tarefa de realização de consultas, ou seja, não têm conhecimentos sobre a tecnologia de banco de dados, sobre modelo de dados, sobre linguagem de consulta e ainda pode haver falta de conhecimento sobre a semântica das informações armazenadas.

Buscando solucionar este problema, estamos propondo neste capítulo uma arquitetura que incorpora uma nova tecnologia aos sistemas de consulta, as interfaces inteligentes. Esta nova tecnologia requer do sistema o armazenamento e manutenção de informações que retratem o conhecimento do usuário sobre o sistema e a tarefa, permitindo a realização de uma forma de cooperação, que é a cooperação do sistema com o usuário.

Este capítulo encontra-se dividido em 4 seções. A primeira seção apresenta uma introdução sobre os objetivos da proposta apresentada e as mudanças em relação aos sistemas atuais. Na segunda seção apresentamos e discutimos as características

funcionais e visuais pretendidas para os sistemas obtidos com esta arquitetura, e que caracterizam a cooperação pretendida. Na seção seguinte apresentamos a arquitetura e detalhamos seus módulos. E finalmente a quarta seção conclui este capítulo.

4.1 Introdução

A evolução dos sistemas de consulta ocorrida desde a década 60, vem demonstrando a preocupação da comunidade de banco de dados em facilitar o processo de definição, mas principalmente de manipulação do banco de dados. Essa evolução tem se apresentado, principalmente, através da variação nos paradigmas visuais utilizados e das facilidades implementadas nos sistemas.

Esses sistemas apresentam como arquitetura básica a apresentada na Figura 4.1. Esta arquitetura caracteriza-se pela existência de três camadas principais: interface, o sistema de consulta visual e o sistema gerenciador de banco de dados.

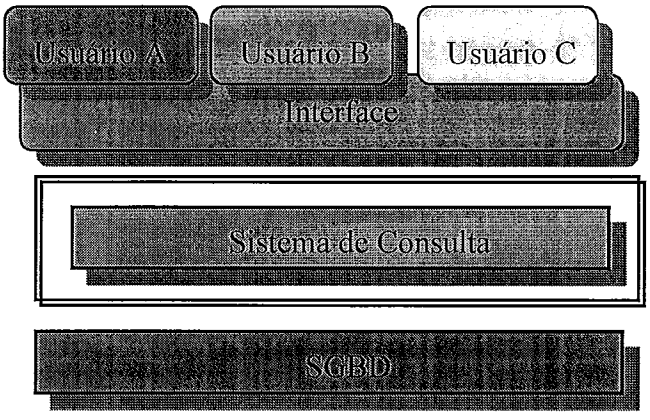


Figura 4.1 - Arquitetura Tradicional

A camada de interface é caracterizada pela existência de uma interface comum a todos os usuários do sistema, gerada pelo sistema de consulta visual. Essa interface padrão permite que os usuários acessem os dados armazenados na base através do SGBD.

Em alguns sistemas, como ocorre no sistema ICONICBROWSER(TSUDA *et al.*, 1990), a interface pode variar de acordo com o esquema da base de dados consultada. Entretanto, essa variação da interface não configura uma nova interface, pois para cada base de dados uma única interface ocorre para todos os usuários.

Os sistemas desenvolvidos, baseados nesta arquitetura, elaboram suas interfaces sem considerar dois aspectos importantes sobre os usuários do sistema:

- *a existência de um público alvo não homogêneo, ou seja, onde é difícil definir características comuns entre os possíveis usuários do sistema, que possam determinar como a interface deve ser construída.*
- *a capacidade de evolução do usuário, ou seja, o usuário pode, à medida que utiliza o sistema, ter suas características alteradas, o que resultaria em novas janelas, possivelmente com outras formas de interação, caso fosse feito um novo projeto de interface.*

Para solucionar esses problemas, estamos propondo uma arquitetura, para sistemas de consultas visuais para banco de dados orientado a objetos, que incorpora ações inteligentes, permitindo uma cooperação da máquina com o usuário e garantindo uma melhora no processo de realização de consultas, principalmente para usuários inexperientes. Especificamente, para solucionar os aspectos considerados acima, a arquitetura utiliza o conceito de interfaces inteligentes, mais especificamente de interfaces adaptativas.

A escolha do modelo orientado a objetos, para a base de dados consultada pelos sistemas desta arquitetura, ocorreu em virtude do menor conhecimento deste modelo de dados pelos usuários e consequentemente maior dificuldade de utilização e pela existência de um menor número de sistemas de consulta desenvolvidos.

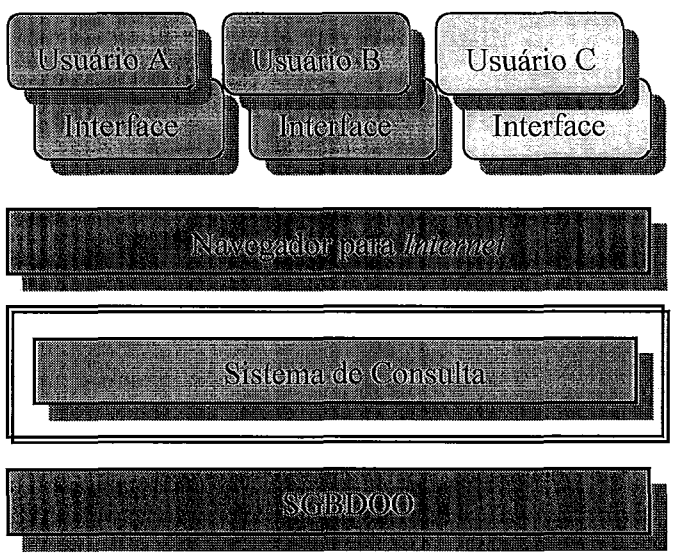


Figura 4.2 - Arquitetura proposta em alto nível.

Em um nível elevado, a arquitetura que estamos propondo pode ser vista como mostra a Figura 4.2. Nesta arquitetura são utilizadas quatro camadas: a camada de

usuários com suas interfaces correspondentes, a camada do navegador da rede mundial de computadores, a camada do sistema de consulta e a camada do sistema gerenciador de banco de dados orientado a objetos(SGBDOO).

Na camada de usuários, representados pelos usuários simbólicos (A, B e C), existe uma interface distinta para cada usuário em questão. A construção desta interface é feita através de decisões tomadas pelo sistema, baseadas no histórico individual de utilização do sistema pelos usuários.

Dessa forma, na fase projeto da interface do sistema não se define apenas uma única interface capaz de atender a todos os possíveis usuários do sistema e sim uma interface básica e variações possíveis para os possíveis tipos de usuários do sistema. E, à medida que o histórico da utilização do sistema pelo usuário é mantido e utilizado para adaptar a interface, podemos garantir que tanto a evolução do usuário, quanto a alteração de suas características serão incorporadas nas interfaces produzidas pelos sistemas desta arquitetura.

O navegador encontra-se entre os usuários e o sistema, como um meio provedor de apresentação das interfaces. Entretanto, a arquitetura proposta é independente da utilização de um navegador. Sua colocação no ambiente, apresentado na Figura 4.2, visa ressaltar que um dos objetivos desta arquitetura é atender aos usuários que acessam bases de dados orientadas a objetos através da rede mundial de computadores e neste caso existe a necessidade do uso de um navegador.

Outra camada apresentada é a do sistema de consultas. Este sistema, desenvolvido de acordo com a arquitetura que será proposta, realiza a comunicação entre o usuário e o sistema gerenciador de banco de dados orientado a objetos(SGBDOO). O sistema de consulta se diferencia dos anteriores pela utilização de conhecimento na construção de suas interfaces e realização das consultas.

4.2 Características de cooperação

Como já citamos, a arquitetura proposta permitirá que o sistema coopere com o usuário, tornando sua tarefa mais simples de ser concretizada. Para alcançar esta cooperação algumas características especiais, chamadas de características de cooperação, foram embutidas na arquitetura. Essas características podem ser separadas em três tópicos: *interface*, *funcionalidades de cooperação* e *geração de consulta.*

A interface dos sistemas é a forma de comunicação com o usuário e por isso corresponde ao meio de cooperação mais imediato dos sistemas. Quanto mais simples e imediata for a interface do sistema, mais rápida será a concretização da tarefa pelo usuário. As formas de interações visuais vêm sendo preteridas pelos desenvolvedores de sistemas, por permitirem essa facilidade e rapidez mencionadas, melhorando a comunicação do sistema e garantindo um primeiro nível de cooperação.

Muitos autores, entre os quais podemos citar HUANG(1988), defendem que o sucesso e a produtividade do usuário na realização de suas tarefas computacionais estão intimamente relacionados à qualidade da interface do sistema que ele utiliza. Por isso, a escolha do paradigma visual tornou-se uma tarefa importante no processo de desenvolvimento de sistemas.

A proposta desta arquitetura está baseada na utilização de paradigmas visuais para todas as interfaces necessárias ao sistema de consulta, de forma que o usuário crie suas consultas utilizando apenas a interação com os elementos visuais apresentados.

Outra característica de cooperação embutida na interface é a adaptação. O sistema procura descobrir que elementos o usuário necessita para realizar suas consultas, bem como, que elementos ele poderá necessitar para consultas posteriores. Essas duas características da interface serão descritas de forma mais detalhada no decorrer desta seção.

Ao realizar o levantamento dos sistemas de consultas visuais existentes na literatura verificamos algumas funcionalidades interessantes, que já promovem um tipo de auxílio ao usuário na realização da tarefa de consulta e no processo de aprendizado do esquema de dados. A arquitetura proposta mantém esse conjunto de funcionalidades:

- *visualização da consulta(SQL ou OQL) gerada*
- *possibilidade de armazenamento de consultas*
- *utilização de consultas salvas como componentes de outras consultas*
- *possibilidade de visualizar subesquemas ou visões do esquema*
- *criação automática dos relacionamentos entre entidades na consulta*
- *apresentação dos objetos relacionados de um objeto no resultado da consulta*

Outra forma de cooperação adotada nesta arquitetura é realizada sobre a consulta visual construída pelo usuário. É possível que esta consulta retorne resultados nulos, que

para usuários inexperientes consiste em uma resposta difícil de ser tratada. Para solucionar este problema, a arquitetura incorpora um processo de relaxamento das consultas.

O relaxamento das consultas pode acarretar mudanças sobre o predicado da consulta(cláusula WHERE) e/ou na coleção origem de dados da consulta(cláusula FROM).

4.2.1 Paradigma Visual

Na maioria das metodologias de desenvolvimento de sistemas, a elaboração da interface do sistema é feita durante a fase de projeto e estas são construídas de acordo com as funções do sistema, sem uma preocupação adequada com o usuário que irá utilizar o sistema.

O tratamento de requisitos não funcionais como desempenho, facilidade de aprendizado, facilidade de utilização, segurança e privacidade não são considerados adequadamente no processo de desenvolvimento de sistemas (PARKER *et al.*,1997).

Os requisitos desempenho(realização da tarefa), facilidade de aprendizado e facilidade de utilização estão intimamente associados ao paradigma de interação visual escolhido para o sistema.

Novas metodologias, baseadas no envolvimento do usuário durante o processo de elaboração das interfaces, estão sendo desenvolvidas. A metodologia apresentada por SUTCLIFFE e MCDERMOTT(1991), por exemplo, propõe a realização do projeto de interface utilizando métricas simples e baseadas em julgamento intuitivo para determinar a sofisticação das habilidades do usuário e o tipo de auxílio a ser fornecido.

Esse tipo de metodologia pode ser aplicado a sistemas com um público alvo bem definido, onde é possível reconhecer os usuários do sistema. Nossa proposta baseia-se no total desconhecimento dos usuários do sistema e por isso é preciso estabelecer formas de interação, ou seja, de comunicação com o usuário, que sejam adequadas a qualquer tipo de usuário, com qualquer tipo de habilidade.

4.2.2 Adaptação

A adaptação de interface proposta nesta arquitetura é realizada entre sessões e em alguns casos, de acordo com as decisões do sistema, durante uma mesma sessão. A maior parte da adaptação é baseada no conteúdo apresentado, ou seja, nos elementos de consulta que serão disponibilizados pelo sistema ao usuário. Essa escolha é feita

baseada nas informações do modelo do usuário, na base de metadados e no conjunto de regras de adaptação.

Elemento de consulta é qualquer componente associado ao esquema de dados da base consultada e/ou à tarefa de realização de consulta, que possa ser utilizado no processo de criação de uma consulta em um modelo orientado a objetos.

Assim, podemos chamar de *elementos da consulta* as classes, atributos e métodos do esquema de dados, bem como os operadores da linguagem de consulta OQL e as consultas armazenadas deste usuário. Exceto o último elemento citado, todos os outros são indispensáveis ao processo de construção de consultas no modelo orientado a objetos.

A adaptação entre seções ocorre na iniciação do sistema de consulta pelo usuário. Neste momento, o sistema descobrirá a partir de inferências à sua base de conhecimento que informações devem ser apresentadas ao usuário em questão. O objetivo principal desta adaptação é criar uma interface que apresente os elementos de consulta "corretos" para o usuário criar suas consultas.

A adaptação durante uma mesma sessão ocorre enquanto o usuário utiliza o sistema de consulta e tem como objetivo principal melhorar a interface de consulta, aumentando ou diminuindo o volume de informações apresentadas, prevendo que elementos de consulta o usuário poderá acessar no futuro próximo bem como que elementos ele não irá mais acessar. Essa forma de adaptação é baseada no nível de aceitação que o usuário demonstrou com relação aos elementos de consultas já apresentados.

Com estas melhorias realizadas na interface, o usuário aumenta de forma gradual o seu conhecimento sobre o esquema de dados consultado e o sistema constrói interfaces de consulta cada vez mais próximas do ideal para o usuário.

É perfeitamente possível e aceitável que o sistema "erre" ao criar a interface do usuário. Como o sistema aprende à medida que é utilizado pelo usuário, as primeiras interfaces produzidas podem não atender aos requisitos desejados pelo usuário, ou seja, elementos de consulta que o usuário deseja utilizar não são apresentados, por isso, é indispensável que a interface permita que o usuário adicione e retire elementos de consulta da interface.

Dentre os elementos de consulta que o sistema seleciona para um determinado usuário, encontram-se também os relacionamentos. Resolvemos realizar uma junção

entre a funcionalidade de criação automática dos relacionamentos e a adaptação. Somente os relacionamentos selecionados para o usuário serão criados automaticamente, os demais devem ser explicitamente estabelecidos pelo usuário.

Outra forma de adaptação menos explorada neste trabalho está baseada na inclusão e remoção de elementos de interface. *Elementos de Interface* são os componentes visuais, como botões, painéis, quadros de texto, caixas de listas entre outros, que podem ser apresentados em uma interface.

4.2.3 Relaxamento da Consulta

Relaxar uma consulta consiste em alterá-la aumentando a gama de resultados possíveis a fim de garantir resultados úteis ao usuário. Situações onde as consultas podem ser modificadas são as seguintes: quando uma consulta retorna um valor nulo e quando uma consulta está errada, ou seja, a consulta visual não pode ser traduzida em uma consulta declarativa.

No primeiro caso, será realizado um relaxamento da consulta buscando aumentar o escopo dos possíveis resultados, fornecendo resultados aproximados ao que foi requisitado. Já no segundo caso, um tratamento de correção da consulta é necessário.

Na literatura encontram-se retratadas alguns formas de implementação de relaxamento de consultas para o modelo de dados relacional. CHU *et. al.*(1993) desenvolveu uma estrutura, chamada de árvore abstrata de tipos, composta por árvores de abstração sobre as relações, sobre tuplas das relações e sobre os valores dos elementos das tuplas. Esta estrutura de dados é utilizada em conjunto com regras que determinam como e quando o relaxamento da consulta deve ser efetuado. Outra proposta é apresentada por RAMOS *et al.*(1996), baseada na utilização de duas bases de conhecimento. Uma base contém informações sobre o domínio da aplicação, as relações, atributos, domínio dos atributos, chaves e outros. A outra base contém as regras para realizar as transformações nas consultas

O relaxamento de consultas é mais um mecanismo de cooperação entre o sistema e o usuário baseado na manutenção de informações de conhecimento. Individualmente esse tipo de cooperação reduz o conhecimento sobre a linguagem de consulta necessário ao usuário e ainda permite um aprendizado do esquema da base de dados consultada, a partir das respostas produzidas.

Em conjunto com as outras características apresentadas, paradigma visual, adaptação da interface e funcionalidades que auxiliam o usuário, podemos obter sistemas que melhoram a realização da tarefa de consulta em banco de dados orientado a objetos por usuários inexperientes.

4.3 Arquitetura

A proposta deste trabalho é definir uma arquitetura para o desenvolvimento de sistemas de consultas visuais que sejam capazes de tratar diversos tipos de conhecimento(usuário e tarefa) para garantir uma cooperação do sistema com o usuário.

Nossa arquitetura, apresentada na Figura 4.3, é baseada em módulos com objetivos específicos. Essa divisão foi necessária em virtude da variedade de *tipos* de informação que precisam ser considerados, informações de conhecimento, de interface e de linguagem de consulta.

Outra virtude obtida com a separação em módulos é melhorar o desenvolvimento e a manutenção dos sistemas e possibilitar a troca de módulos, em caso de necessidade de alteração. Estes módulos se comunicam através de protocolos pré-estabelecidos.

Para um melhor entendimento da arquitetura a ser proposta, torna-se necessário apresentar algumas associações realizadas na Figura 4.3 entre as formas gráficas utilizadas (representação) e a denominação utilizada para referenciar estes elementos(significado). Assim, os elementos representados pelo retângulo cheio correspondem aos módulos da arquitetura, que podem participar de um agrupamento representado por um retângulo maior. A linha tracejada com pontos representa uma camada da arquitetura, a linha tracejada com segmentos indica um agrupamento de elementos que não são módulos, as setas indicam a troca de mensagens entre os elementos, módulos ou camadas, através do uso de um protocolo de comunicação, e os elementos ovais indicam componentes fora do nosso escopo de desenvolvimento, ou seja, "pré-fabricados" externamente e necessários à arquitetura.

As camadas foram utilizadas como forma de facilitar a apresentação e o entendimento da arquitetura. Elas agrupam módulos que se comunicam com as mesmas camadas e que possuem funcionalidades semelhantes. Assim, os módulos que de alguma forma interagem com a interface foram agrupados em uma camada chamada *Camada de Interface*. Os módulos que lidam com o conhecimento e tomam decisões

que interferem na tarefa do usuário estão agrupados em uma camada chamada *Camada de Conhecimento* e finalmente os módulos que fazem a conversação com os componentes externos à interface, a máquina de inferência e o SGBD, estão agrupadas na camada chamada de *Camada de Tradutores*.

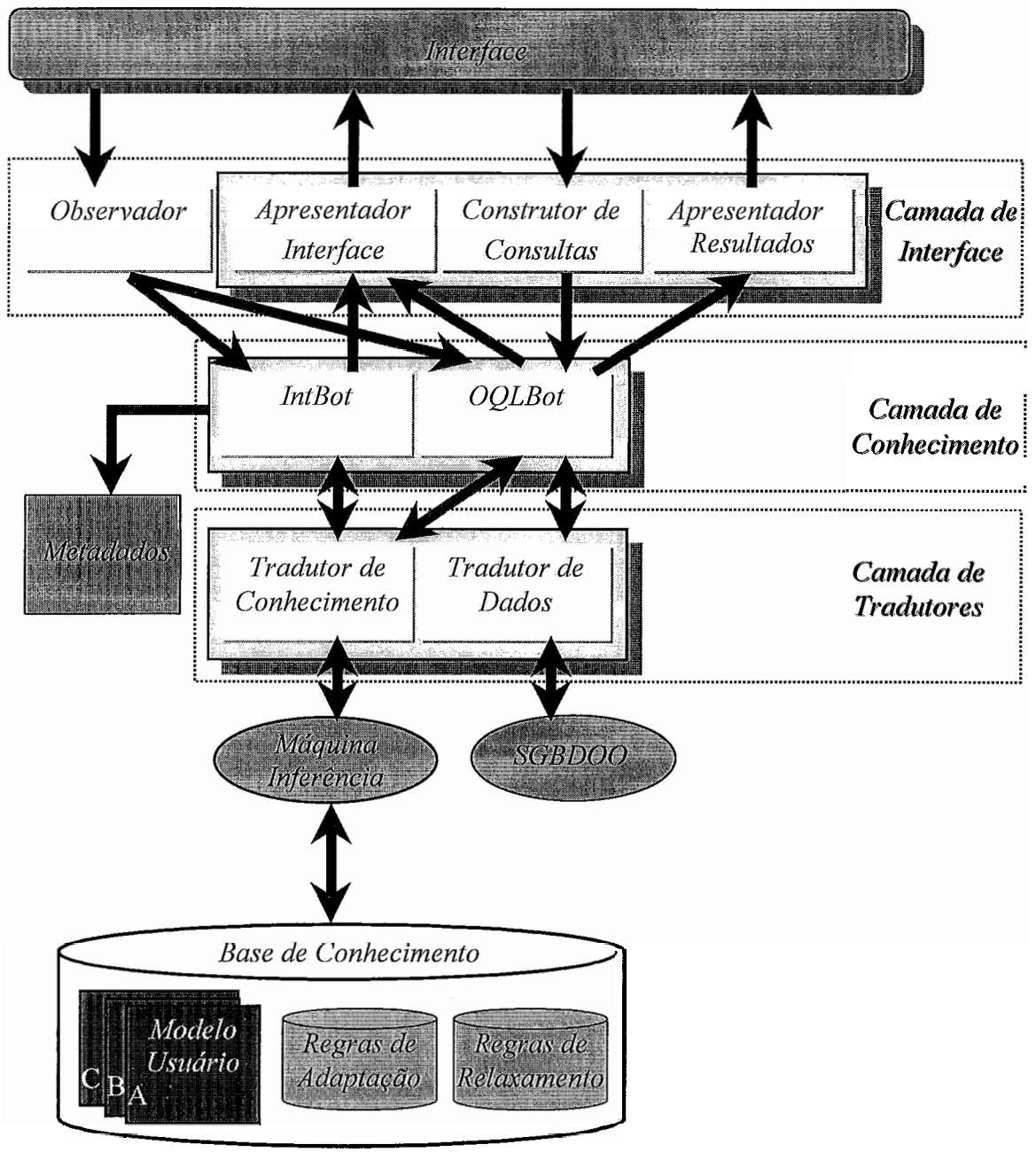


Figura 4.3 - Arquitetura para Sistemas de Consultas Visuais com Interfaces Inteligentes.

Na *Camada de Interface* existe um módulo isolado chamado de *Observador* e um agrupamento de módulos que contém *Apresentador de Interface, Construtor de Consultas* e o *Apresentador de Resultados.* O módulo *Observador* não participa do

agrupamento, pois ao contrário dos demais, não interfere diretamente na interface, atuando como um vigia das ações do usuário na interface e informando-as à *Camada de Conhecimento.*

Na *Camada de Tradutores* torna-se mais visível a vantagem do desenvolvimento da arquitetura em módulos, pois caso ocorra alguma alteração nos elementos externos, somente os módulos desta camada precisam ser alterados ou especializados.

4.3.1 Camada de Interface

Os módulos desta camada são responsáveis pelos processos de criação e manutenção da interface inteligente de consulta. A maioria dos módulos desta camada atua sobre a interface modificando-a, incluindo ou removendo representações visuais dos elementos de consulta ou elementos de interface.

4.3.1.1 Módulo Observador

Este módulo é responsável por observar as interações do usuário com o sistema. Essas interações são transformadas em informações e passadas para a camada de conhecimento que irá processá-las e determinará as ações a serem realizadas. Parte dessas informações são utilizadas para construção e alteração do modelo do usuário.

Interações relativas à interface do sistema de consulta são capturadas por este módulo com exceção das interações realizadas para construção da consulta.

A Tabela 4.1 apresenta o conjunto de mensagens que este módulo é capaz de tratar. Podemos perceber que todas têm origem na interface, ou seja, na interação do usuário com o sistema. Essas mensagens reproduzem as funcionalidades que o sistema deve apresentar para garantir as características de cooperação descritas anteriormente.

Uma das características de adaptação da interface é baseada na seleção de elementos de consulta que serão apresentados aos usuários. Entretanto, principalmente nas primeiras interações do usuário com o sistema, essa seleção pode não atender às necessidades do usuário, por isso, é necessário permitir a adição e remoção de elementos de consulta à interface.

Origem	Mensagem	Parâmetros de Entrada
Interface	Informar elementos de consulta para adição(classe, método, atributo, operador, relacionamento)	Elementos de consulta apresentados, Tipo do elemento de consulta, Usuário
Interface	Adicionar elemento de consulta(classe, método, atributo, operador, relacionamento)	Elemento de consulta, Tipo do elemento de consulta, Usuário
Interface	Retirar elemento de consulta(classe, método, atributo, operador, relacionamento)	Elemento de consulta, Tipo do elemento de consulta, Usuário
Interface	Informar lista de consultas armazenadas	Usuário
Interface	Pedido para armazenar consulta	
Interface	Armazenar consulta	Nome da consulta, Consulta, Usuário
Interface	Navegar pelo Relacionamento	Objeto apresentado, relacionamento
Interface	Finalizar interface	

Tabela 4.1 - Protocolo entrada do Módulo Observador.

4.3.1.2 Módulo Apresentador Interface

Este módulo é responsável por atender as requisições de alteração da interface do usuário provenientes da camada de conhecimento. É sua função decidir, de acordo com o paradigma visual adotado, como os elementos de consulta devem ser apresentados, ou seja que elemento visual deve ser utilizado em correspondência ao elemento de consulta que foi determinado para apresentação e em que local deve ser inserido. Ao mesmo tempo, em caso de retirada de elementos, é função deste módulo reorganizar a interface. Em ambos os casos, deve procurar mantê-la amigável e de fácil utilização para o usuário.

Este módulo trata as mensagens apresentadas na Tabela 4.2.

Origem	Mensagem	Parâmetros de Entrada
IntBot	Apresentar elemento de consulta(classe, método, atributo, operador, relacionamento)	Elemento de consulta, Tipo do elemento de consulta
IntBot	Apresentar lista de elementos de consulta(classe, método, atributo, operador, relacionamento, consultas armazenadas)	Lista de elementos de consulta, Tipo dos elementos de consulta
IntBot	Retirar elemento de consulta(classe, método, atributo, operador, relacionamento)	Elemento de consulta, Tipo do elemento de consulta
IntBot	Apresentar lista de consultas armazenadas	Lista dos nomes consultas salvas, Lista da oql das consultas salvas
IntBot	Adicionar consulta armazenada	Nome da consulta,Oql da consulta
IntBot	Apresentar elemento de interface	Elemento de Interface
IntBot	Adicionar elemento de interface	Elemento de Interface
IntBot	Remover elemento de interface	Elemento de Interface
IntBot	Apresentar auxílio	Texto de auxílio
IntBot	Finalizar apresentação	
OQLBot	Apresentar consulta linguagem declarativa	Nome da consulta, Oql da consulta
OQLBot	Apresentar janela para troca de parâmetros de consulta armazenada	Número de parâmetros, Lista de nomes dos parâmetros, Lista de tipos dos parâmetros

Tabela 4.2 - Protocolo entrada do Módulo Apresentador de Interface.

Todas essas mensagens de entrada no módulo apresentador de interface geram um mensagem de saída associada, destinada à interface. Essas mensagens de saída determinam a realização efetiva da operação determinada na mensagem de entrada.

Por exemplo, para a mensagem *Apresentar elemento de consulta,* após o processamento efetuado neste módulo, uma mensagem de saída é gerada e enviada a interface determinando a apresentação do um elemento visual correspondente ao elemento de consulta.

4.3.1.3 Módulo Apresentador Resultados

A consulta construída pelo usuário é enviada à camada de conhecimento e processada pelo SGBDOO. As respostas produzidas por esta consulta são tratados pela camada de conhecimento e quando este processo estiver completo os resultados são enviados à camada de interface para serem apresentados.

Na camada de interface, o módulo apresentador de resultados organiza e apresenta ao usuário os resultados obtidos pela consulta. Este módulo determina o paradigma visual a ser utilizado e a forma de apresentação, garantindo a navegação pelos relacionamentos dos objetos recuperados.

A Tabela 4.3 apresenta as mensagens tratadas por este módulo :

Origem	Mensagem	Parâmetros de Entrada
OQLBot	Inicia apresentação de resultados	Título da Janela
OQLBot	Apresentar valor	Nome do atributo, Valor do atributo, Identificador do objeto que contém este valor
OQLBot	Apresentar lista de valores	Nome do atributo, Identificador do objeto que contém esta lista
OQLBot	Apresentar lista de relacionamentos	Nome do atributo, Identificador do objeto que contém esta lista
OQLBot	Apresentar relacionamento	Nome do Relacionamento, Identificador do objeto que contém este relacionamento
OQLBot	Apresentar Elemento de Interface	Elemento de Interface
OQLBot	Finalizar apresentação	

Tabela 4.3 - Protocolo entrada do Módulo Apresentador de Resultados.

Do mesmo modo como acontece no módulo apresentador de interface, mensagens de saída associadas às mensagens de entrada do módulo apresentador de resultados são geradas para a interface.

Nesta arquitetura não fixamos nenhum paradigma visual de apresentação dos resultados, estabelecendo somente a navegação entre os relacionamentos recuperados na consulta realizada.

4.3.1.4 Módulo Construtor de Consultas

Neste módulo da Camada de Interface são construídas as consultas visuais, baseadas na organização de elementos visuais, e capturadas as interações do usuário com o sistema relativas à consulta, que são: o processo de construção da consulta visual, visualização da consulta declarativa gerada(funcionalidade cooperativa) e alteração de parâmetros da consulta.

Para cada base de dados disponibilizada pelo sistema, um dicionário de elementos visuais tem que ser criado. Estes dicionário contém a representação visual que será utilizada para todos os elementos de consulta dos tipos: classes, atributos, métodos e operadores.

A combinação desses elementos visuais, de acordo com as regras estabelecidas neste módulo, serão traduzidas em consultas válidas para o modelo orientado a objetos. Podemos dizer que esta combinação gera uma sentença de elementos visuais.

As regras para a criação de sentenças válidas serão apresentadas a seguir. Essas regras estão separadas pelos tipos de elementos de consulta que podem ser adicionados a uma sentença:

- *Adição de classes*

  As classes podem ser adicionadas às sentenças com dois objetivos distintos:

  - estabelecer a coleção origem dos dados

    O primeiro elemento a ser adicionado a uma sentença deve ser uma classe. A coleção associada a esta classe será utilizada como fonte de dados para a operação da consulta.

  - realizar navegação

Quando o objetivo é realizar uma navegação, a classe pode ser adicionada à sentença e em seguida o relacionamento que permite alcançar esta classe será criado automaticamente pelo sistema ou adicionado pelo usuário.

- realizar uma junção

  Para realizar uma junção é preciso adicionar a classe em seguida a um operador de comparação(>, <, >=, <=, =, !=), construindo o predicado.

➲ *Adição de relacionamentos*

Os relacionamentos podem ser adicionados à sentença desde que a classe origem e a classe destino do relacionamento já componham a sentença. Através da inserção dos relacionamentos, garante-se o processo de navegação, descrito na linguagem OQL. Além disso, permite acessar novos elementos de consulta(atributos, metodos e classes) que podem ser utilizados na construção do predicado da consulta.

➲ *Adição de atributos e ou métodos*

Os atributos e métodos de uma classe podem ser adicionados à sentença desde que a classe correspondente já tenha sido também adicionada à sentença.

➲ *Adição de operadores*

Os operadores podem ser adicionados em seguida à uma classe, atributo, método, consulta armazenada ou outro operador. Esta regra necessita de uma especialização de acordo com o tipo dos operadores, pois nem todos os operadores se aplicam a todos os tipos de elementos de consulta. Por exemplo, não constituí uma regra válida adicionar um operador de comparação seguido a outro operador de comparação. A seguir vamos apresentar as regras para cada tipo de operador individualmente.

- Operadores de comparação(>, <, >=, <=, =, !=)

  Devem ser aplicados entre elementos de consulta do mesmo tipo(numérico ou cadeia de caracteres) quando se deseja obter valores

do tipo verdadeiro ou falso. São utilizados para compor os predicados das consultas.

- Operadores de agregação(sum, avg, count, min, max)

  Devem ser aplicados precedendo elementos de consulta que retornem conjuntos de tipo numérico, com exceção do operador count que deve ser aplicado antes de elementos que representem conjunto de elementos, que pode ser um atributo, método, consulta armazenada ou a própria consulta que está sendo construída. Neste caso, deve ser inserido no início da sentença.

- Operadores de coleção(first, last, in)

  Devem ser aplicados precedendo elementos de consulta que retornem conjuntos.

- Operadores de agrupamento(group by)

  Deve ser aplicado após um elemento que retorne um valor atômico. A coleção de elementos que será agrupada é aquela que contém o elemento com valor atômico.

- Operadores de ordenação(order by)

  Deve ser aplicado após um elemento que retorne um valor atômico. A coleção de elementos que será ordenada é aquela que contém o elemento com valor atômico.

Ao garantir essas regras de construção de sentenças visuais, o módulo Construtor de Consultas garante a criação de consultas sintaticamente corretas pelo usuário, realizando desta forma uma cooperação implícita neste processo.

Para verificar as restrições de cada regra de criação de sentenças, o módulo requisita informações sobre o esquema de dados consultado à camada de conhecimento, como pode ser visto pelas mensagens do OQLBot provenientes deste módulo.

Além de garantir a correta construção das sentenças visuais, este módulo também atende as seguintes mensagens de entrada descritas na Tabela 4.4 a seguir:

Origem	Mensagem	Parâmetros de Entrada
Interface	Elemento de consulta inserido na consulta(classe, atributo, método, operador, relacionamento, consulta armazenada)	Elemento de consulta, Tipo do elemento de consulta, Posicionamento na sentença visual
Interface	Elemento de consulta retirado da consulta(classe, atributo, método, operador, relacionamento, consulta armazenada)	Elemento de consulta, Tipo do elemento de consulta, Posicionamento na sentença visual
Interface	Elemento de consulta movido(classe, atributo, método, operador, consulta armazenada)	Elemento de consulta, Tipo do elemento de consulta, Posicionamento na sentença visual
Interface	Compor expressão de consulta	
Interface	Elemento de consulta utilizado em consulta	Elemento de consulta, Tipo do elemento de consulta
Interface	Consulta realizada	
Interface	Informar consulta corrente em linguagem declarativa	
Interface	Alterar parâmetros de consulta armazenada	Nome da consulta

Tabela 4.4 - Protocolo entrada do Módulo Construtor de Consultas.

4.3.2 Camada de Conhecimento

Os módulos desta camada utilizam em sua denominação o sufixo *Bot*, definido por A-KAUTZ *et. al.* (1994) como *Software Robot* para utilização em sistemas que apresentem algum tipo de inteligência para tomada de suas decisões e auxílio ao usuário. Esses módulos apresentam características de agentes inteligentes.

4.3.2.1 IntBot

Este módulo é responsável pela tomada de decisões relativas à construção das interfaces. Ele apresenta uma interação muito forte com a máquina de inferência através da camada de tradutores. A partir das respostas produzidas pela máquina de inferência e obtidas por este módulo, pela camada de tradutores, é possível determinar que elementos de consulta devem ser apresentados pela camada de interface.

A adaptação proposta nesta arquitetura pode ocorrer em duas etapas distintas: entre seções distintas de uso do sistema pelo usuário ou durante uma mesma sessão. Em ambos os casos, a máquina de inferência é requisitada por este módulo e através de suas inferências, baseadas nos dados do modelo do usuário, nas informações da base de metadados e nas regras de adaptação, é possível determinar como a adaptação deve ser realizada.

A adaptação entre seções distintas é caracterizada pela construção de uma nova interface, enquanto a adaptação em uma mesma sessão é caracterizada pela modificação de uma interface já apresentada ao usuário. Duas formas de adaptação durante uma mesma sessão foram definidas nesta arquitetura: através de faixas de adição e por proximidade.

Para cada esquema de dados são definidas "n" faixas de valores(<min_1,max_1>, <min_2,max_2>, ..., <min_n,max_n>) não disjuntas e o usuário terá "n" níveis de adição(1,2,3,...,n). A faixa de valores e o nível de adição do usuário possuem uma associação, de forma que o nível de adição 1 corresponde a faixa de valores <min_1,max_1>, o nível de adição 2 corresponde a faixa de valores <min_2,max_2>, e assim sucessivamente.

Cada elemento de consulta é classificado com um valor, chamado faixa de adição, que pertence a uma dessas faixas de valores. Quando o *IntBot* determinar a adaptação da interface por faixa de adição, serão enviados para à camada de interface os elementos de consulta que apresentam um valor de faixa de adição que esteja entre a variação de valores possíveis, correspondente ao nível do usuário.

De acordo com a aceitação dos elementos adicionados, o nível de adição do usuário pode aumentar ou diminuir. Em caso de aumento do nível de adição, novos elementos de consulta são adicionados à interface, no caso de redução do nível de adição do usuário, os elementos de consulta correspondentes a faixa de valores associada ao nível são retirados da interface.

Outra forma de adicionar novas classes à interface é através da adição de classes próximas às classes mais utilizadas. Para esta adaptação, são definidos conjunto de proximidades, ou seja, conjuntos de classes consideradas próximas à classe avaliada, para cada classe do esquema. Esses conjuntos são ordenados para que possam ser adicionados à interface do usuário de forma gradual, minimizando as desvantagens da adaptação durante uma mesma sessão, citadas no capítulo anterior. Novamente, de acordo com a aceitação do usuário, este processo pode avançar ou regredir.

Este módulo trata as seguintes mensagens apresentadas na Tabela 4.5.

Origem	**Mensagem**	**Parâmetros de Entrada**
Observador	Informar elementos de consulta para adição(classe, método, atributo, operador, relacionamento)	Elementos de consulta apresentados, Tipo do elemento de consulta, Usuário, Nome base de dados
Observador	Retirar elemento de consulta(classe, método, atributo, operador, relacionamento)	Elemento de consulta, Tipo do elemento de consulta, Usuário, Nome base de dados
Observador	Informar lista consultas armazenadas	Usuário, Nome base de dados
Observador	Armazenar consulta	Nome da consulta, Consulta, Usuário, Nome base de dados
Observador	Finalizar interface	Usuário, Nome base de dados
Tradutor Conhecimento	Elementos(de consulta e de interface) a serem apresentados	Lista elementos(de consulta e de interface)
Tradutor Conhecimento	Elementos(de consulta e de interface) a serem removidos da interface	Lista elementos(de consulta e de interface)
Tradutor Conhecimento	Elementos(de consulta e de interface) a serem adicionados à interface	Lista elementos(de consulta e de interface)

Tabela 4.5 - Protocolo entrada do *IntBot.*

4.3.2.2 OQLBot

A tendência é que todos os SGBDOO utilizem a OQL(CATTEL,1997) como linguagem de consulta, por isso adotamos o termo OQL para nomear este módulo.

Este módulo é responsável por transformar a sentença de elementos visuais que o usuário constrói na camada de interface em uma consulta em linguagem declarativa. Na realidade, este módulo particiona a sentença de elementos visuais nas cláusulas de uma consulta OQL, obtém as informações necessárias a cada cláusula e envia para o Tradutor de Dados construir a consulta, de acordo com a linguagem de consulta utilizada no SGBDOO. As variações que podem ocorrer entre as linguagens são pequenas, normalmente as cláusulas se mantêm, sendo organizadas de forma diferente da utilizada na OQL e/ou com palavras chaves um pouco distintas.

Caso a consulta construída a partir da sentença de elementos visuais produza resultados vazios, o relaxamento da consulta será efetuado. Essa alteração pode ser realizada em duas etapas: no predicado da consulta(cláusula WHERE) ou na coleção de busca dos dados(cláusula FROM).

No primeiro caso, informações armazenadas na base de metadados são utilizadas. Os atributos de todas as classes do esquema de dados são agrupados, constituindo o conceito de tópico(RAMOS *et. al.*, 1996). Em cada tópico são definidos grupos de valores possíveis de ocorrer no atributo, com uma semântica associada. Através desses grupos é possível realizar o relaxamento do predicado.

Como exemplo, vamos supor uma classe Vôo com um atributo hora de saída. É possível criar para este atributo um tópico chamado período. Neste tópico, quatro grupos distintos de períodos são criados, manhã, tarde, noite e madrugada e os possíveis valores são definidos para cada grupo. Assim, para o grupo manhã podemos definir que o atributo pode variar entre seis horas e meio dia, para tarde entre meio-dia e dezoito horas, noite entre dezoito horas e meia-noite e madrugada entre meia-noite e seis horas.

Assim, suponha que o predicado Vôo.hora de saída > 9:00 esteja sendo relaxado. Inicialmente deve-se obter o tópico associado ao atributo, e em seguida o grupo em que o valor utilizado se enquadra. O predicado então é alterado para Vôo.hora de saída > 6:00 e Vôo.hora de saída < 12:00 de forma que, não só os vôos que partem após às nove horas, mas sim durante todo o período da manhã sejam considerados pela consulta.

A segunda possibilidade de relaxamento de consulta é efetuada utilizando-se a hierarquia de classes do esquema de dados consultado. Estas informações são obtidas do SGBD, através da camada de tradutores. Esse tipo de relaxamento é um pouco mais difícil de se concretizar, pois é preciso que nenhum atributo da classe mais especializada

seja utilizado na consulta. Para apresentar nosso exemplo, suponha a hierarquia de classes apresentada na Figura 4.4.

Baseado nesta hierarquia, vamos supor que o usuário construísse uma consulta para selecionar os vôos com local de saída igual "Rio de Janeiro" e local de chegada igual "Búzios" . Ao submeter a consulta à base de dados, o resultado obtido é vazio, e o processo de relaxamento da consulta será disparado pelo módulo *OQLBot*. Vamos supor então que o módulo ao realizar consultas as suas regras de relaxamento, decidisse aplicar a segunda forma de relaxamento, baseada na hierarquia de classes.

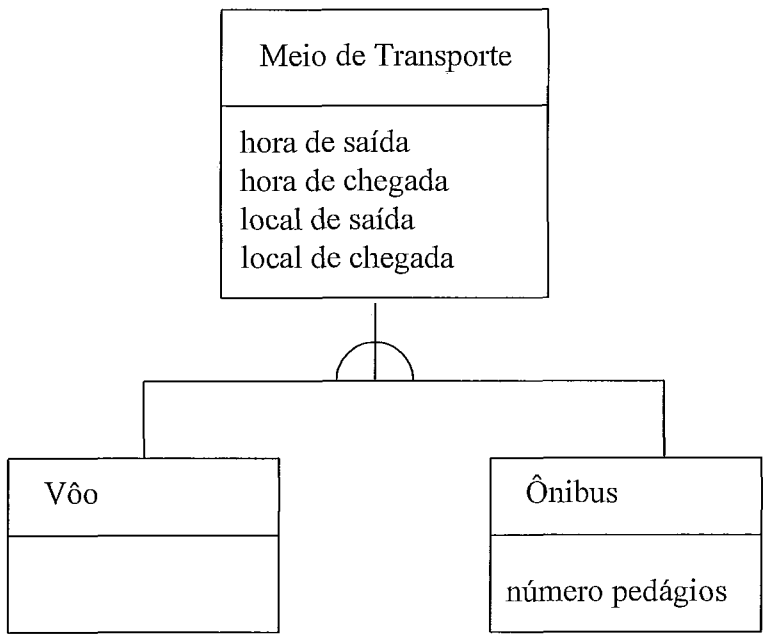


Figura 4.4 - Hierarquia de classes

Ao analisar a hierarquia de classes apresentada na Figura 4.4 podemos verificar que existe uma classe superior à classe utilizada na consulta e que todos os atributos utilizados na consulta pertencem a esta classe superior, portanto é possível modificar a consulta para que a origem dos dados seja a coleção de elementos da classe superior, ficando a consulta da seguinte forma: selecionar os meios de transporte com local de saída igual "Rio de Janeiro" e local de chegada igual "Búzios".

Ao executar esta consulta é possível que o resultado não seja mais nulo, pois além dos vôos, os ônibus também serão verificados com relação ao predicado da consulta.

Dessa forma, a partir desses exemplos, é possível verificar que existem casos em que o relaxamento fornece informações importantes ao usuário, sem que ele as tenha requisitado diretamente evitando a construção de uma nova consulta.

O processo de relaxamento é guiado pelo conjunto de regras de relaxamento armazenadas na base de conhecimento.

A Tabela 4.6 apresenta as mensagens tratadas por este módulo:

Origem	**Mensagem**	**Parâmetros de Entrada**
Construtor de Consultas	Elemento de consulta utilizado em consulta	Nome base de dados, Elemento de Consulta, Tipo do Elemento de Consulta, Usuário
Construtor de Consultas	Profundidade da consulta	Número máximo de navegações em sequência, Usuário, Nome base de dados
Construtor de Consultas	Abertura da consulta	Número máximo de navegações a partir de uma mesma classe, Usuário, Nome base de dados
Construtor de Consultas	Consulta realizada	Seqüência de elementos de consulta, Usuário, Nome base de dados
Construtor de Consultas	Informa consulta em linguagem declarativa	Seqüência de elementos de consulta, Nome base de dados
Construtor de Consultas	Informa existência de relacionamento entre classes	Nome classe origem, Nome classe destino, Nome base de dados
Construtor de Consultas	Informa se atributo ou método pertence à classe	Nome do atributo ou método, Nome da classe, Nome base de dados

Construtor de Consultas	Informa tipo de entrada e saída do método	Nome do método, Nome base de dados
Construtor de Consultas	Informa tipo do atributo	Nome do atributo, Nome base de dados
Construtor de Consultas	Informa tipo do operador	Nome do operador, Nome base de dados
Tradutor de Dados	Resultado da consulta	Lista de objetos resultado
Observador	Navegar por relacionamento de objeto	Identificador do objeto, Nome do relacionamento
Tradutor de Dados	Resposta à consulta realizada	Lista de objetos de resposta
Tradutor de Dados	Objetos relacionados	Lista de objetos relacionados, Nome do relacionamento
Tradutor de Dados	Relacionamento entre classes	Nome relacionamento
Tradutor de Dados	Atributo ou método pertence à classe	Nome atributo ou método, Resposta (sim ou não)
Tradutor de Dados	Tipo de entrada e saída do método	Nome do método, Tipo
Tradutor de Dados	Tipo do atributo	Nome do atributo, Tipo
Tradutor de Dados	Tipo do operador	Nome do operador, Tipo
Tradutor de Dados	Oql da consulta	Nome da consulta Oql da consulta

Tabela 4.6 - Protocolo entrada do *OQLBot.*

As três primeiras mensagens fornecem informações sobre a consulta realizada pelo usuário. Essas informações se transformarão em dados de conhecimento e serão

armazenados no modelo do usuário. A quarta mensagem informa ao módulo *OQLBot* a seqüência de elementos visuais da consulta criada pelo usuário. As demais mensagens tem como funcionalidade garantir as regras de construção de sentença de elementos visuais do módulo Construtor de Consultas. Essas mensagens são passadas ao Tradutor de Dados para que suas informações sejam requisitadas ao SGBD.

4.3.3 Camada de Tradutores

Esta camada é responsável por realizar a comunicação da camada de conhecimento com os componentes "externos" ao sistema, a máquina de inferência e o SGBD.

4.3.3.1 Tradutor de Conhecimento

Este módulo é responsável diretamente pela comunicação com a máquina de inferência, realizando a tradução das perguntas enviadas pela camada de conhecimento em inferências compreendidas pela máquina utilizada.

Este módulo é capaz de compreender as mensagens apresentadas na Tabela 4.7.

Origem	Mensagem	Parâmetros de Entrada
IntBot	Iniciar máquina de inferência	
IntBot	Carregar base de regras de adaptação	Base de regras de adaptação
IntBot	Carregar fatos do usuário	Base de fatos do usuário
IntBot	Inferir a máquina de inferência	Pergunta a ser realizada
IntBot	Definir fato novo	Fato a ser adicionado
IntBot	Definir regra nova	Regra a ser adicionada
OQLBot	Carregar base de regras de relaxamento	Nome da base de regras de relaxamento
OQLBot	Inferir a máquina de inferência	Pergunta a ser realizada
OQLBot	Definir fato novo	Fato a ser adicionado
OQLBot	Definir regra nova	Regra a ser adicionada
Máquina de Inferência	Resposta à inferência	Lista de elementos(Elementos de Consulta e de Interface)

Tabela 4.7 - Protocolo entrada do Tradutor de Conhecimento.

Na mensagem "Inferir a máquina de inferência" estão subentendidas as diversas perguntas determinadas pela módulo *IntBot* para determinar a construção da interface de consulta do usuário, como: Que elementos(de consulta e de interface) devem ser apresentados ?, Que elementos(de consulta e de interface) devem ser retirados da interface ? e ainda Que elementos(de consulta e de interface) devem ser adicionados à interface ?

4.3.3.2 Tradutor de Dados

Este módulo trata as mensagens apresentadas na Tabela 4.8.

Origem	Mensagem	Parâmetros de Entrada
OQLBot	Informar hierarquia de classes	Nome base de dados
OQLBot	Informar atributos de classe	Nome base de dados, Nome da classe
OQLBot	Informar métodos de classe	Nome base de dados, Nome da classe
OQLBot	Constrói consulta	Seqüência de elementos de consulta
OQLBot	Realiza consulta	Nome base de dados, Oql da consulta
OQLBot	Informar relacionamento de objeto	Nome base de dados, Nome do relacionamento, Identificador do objeto,
OQLBot	Informa existência de relacionamento entre classes	Nome base de dados, Classe origem, Classe destino
OQLBot	Informa se atributo pertence à classe	Nome base de dados, Nome do atributo, Nome da classe
OQLBot	Informa se método pertence à classe	Nome base de dados, Nome do método, Nome da classe
OQLBot	Informa tipo de saída do método	Nome base de dados, Nome da classe, Nome do método
OQLBot	Informa tipo de entrada do método	Nome base de dados, Nome da classe, Nome do método
OQLBot	Informa tipo do atributo	Nome base de dados, Nome da classe, Nome do atributo
OQLBot	Informa tipo do operador	Nome do operador

Tabela 4.8 - Protocolo entrada do Tradutor de Dados.

Sua função é realizar a mediação de dados, operando como interlocutor entre a camada de conhecimento e o SGBDOO.

As três primeiras mensagens são utilizadas para a obtenção de informações sobre o esquema da base de dados consultada para que o módulo *OQLBot* realize o relaxamento da consulta. As duas seguintes estão relacionadas à consulta criada pelo usuário e a sexta mensagem é relacionada à apresentação dos relacionamentos que compõem o resultado de uma consulta. As demais mensagens são para a construção de sentenças visuais sintaticamente corretas.

4.3.4 Máquina de Inferência

Este componente faz parte da arquitetura básica dos agentes inteligentes e tem como funcionalidade obter respostas às questões enviadas pela Camada de Tradutores, através da manipulação dos dados da base de conhecimento.

4.3.5 Modelo do Usuário

Contém as informações individuais de cada usuário, permitindo desta forma a tomada de decisões na adaptação da interface e no relaxamento das consultas. Para determinar que tipo de informações são relevantes ao processo de realização de consultas foram avaliados os trabalhos de CATARCI *et. al.*(1991) e MAYHEW(1992) e elucidados os seguintes fatores :

- *importância da tarefa para o usuário*
- *tipo de acesso do usuário(opcional ou obrigatório)*
- *experiência do usuário na tarefa*
- *experiência do usuário no sistema*
- *freqüência de acesso do usuário*
- *tipo de consulta realizada pelo usuário(repetitivas ou extemporâneas)*
- *complexidade da consulta*

Avaliando a tarefa de realização de consulta, sentimos a necessidade de adicionar outras informações sobre a tarefa realizada pelo usuário. Essas informações apresentam um histórico de utilização do sistema e também de acesso a base de dados, possibilitando que o sistema decida que informações devem ser disponibilizadas ao usuário.

Além desses fatores, alguns outros foram adicionados:

- *número médio de operadores utilizados*
- *número médio de classes utilizadas*
- *número de consultas realizadas*
- *número de acessos realizados*
- *classes, operadores e relacionamentos usados pelo usuário*
- *as alterações que o usuário processou na interface*
- *as alterações que o sistema processou na interface*
- *momento no tempo em que as alterações se realizaram*
- *a aceitação das alterações de interface que o sistema realiza para o usuário*

O modelo de usuário desta arquitetura é individual, evolutivo, restrito à tarefa de realização de consultas, tendo suas informações armazenadas como números e pesos.

4.3.6 Regras de Adaptação

As regras de adaptação são utilizadas pela máquina de inferência para decidir o que deve ser apresentado ao usuário em sua interface, tanto ao iniciar a sessão do sistema de consulta quanto durante uma sessão. Utilizando a base de regras em conjunto com o modelo do usuário e os metadados é possível responder às inferências apresentadas na Tabela 4.9.

Inferências
Que elementos de consulta(classe, atributo, método, operador e relacionamento) devem ser apresentados na interface
Que elementos de consulta(classe, atributo, método, operador e relacionamento) devem ser adicionados à interface
Que elementos de consulta(classe, atributo, método, operador e relacionamento) devem ser retirados da interface
Que elementos de interface devem ser apresentados na interface
Que elementos de interface devem ser retirados da interface

Tabela 4.9 - Conjunto de inferências respondidas com o uso da Base de Regras de Adaptação

A construção da base de regras para adaptação é feita de forma livre pelos sistemas implementados de acordo com esta arquitetura. Para tornar viável a utilização das regras de adaptação é importante que elas sejam independentes do esquema de dados utilizado.

Assim, propomos a definição de conceitos, que possam ser utilizados para caracterizar os elementos de consulta, independente do esquema de dados. Assim, a partir de uma pré-classificação dos elementos de consulta, de acordo com esses conceitos, a máquina de inferência será capaz de informar quais elementos devem ser apresentados. Essa pré-classificação criará grupos de elementos de consulta, onde os elementos de um mesmo grupo apresentem afinidades com relação aos usuários deste esquema de dados.

Caso seja indispensável a criação de regras dependentes do esquema de dados, sugerimos a criação de uma segunda base de regras de adaptação, que será utilizada pela máquina de inferência somente na realização de consultas a base de dados correspondente.

Tanto as regras para adaptação da interface entre sessões distintas, quanto durante uma mesma sessão devem ser baseadas na existência desses conceitos.

4.3.7 Regras de Relaxamento

As regras de relaxamento são utilizadas pelo *OQLBot* através da máquina de inferência para determinar como deve ser realizado o processo de relaxamento de uma determinada consulta. O conjunto de dados que abastece essas regras são a hierarquia de classes do esquema consultado e a base de metadados.

As inferências realizadas sobre esta base de regras são apresentadas na Tabela 4.10 a seguir:

Inferências
Que atributo deve ser relaxado
Que coleção deve ser relaxada
Informa o tópico do atributo
Informa tópico do tópico

Tabela 4.10 - Conjunto de inferências respondidas com o uso da Base de Regras de Relaxamento.

4.3.8 Metadados

Esta base de dados armazena dados sobre o esquema de dados a ser consultado e é utilizada pelos dois módulos da Camada de Conhecimento para realização de suas tarefas inteligentes. Ao contrário do modelo do usuário, os dados contidos nesta base não são alterados pelo sistema.

O profissional responsável pela construção da base de metadados deve possuir bons conhecimentos na área de banco de dados orientado a objetos e também no esquema da base de dados que será consultado. Para cada esquema de dados, uma base de metadados deve ser construída.

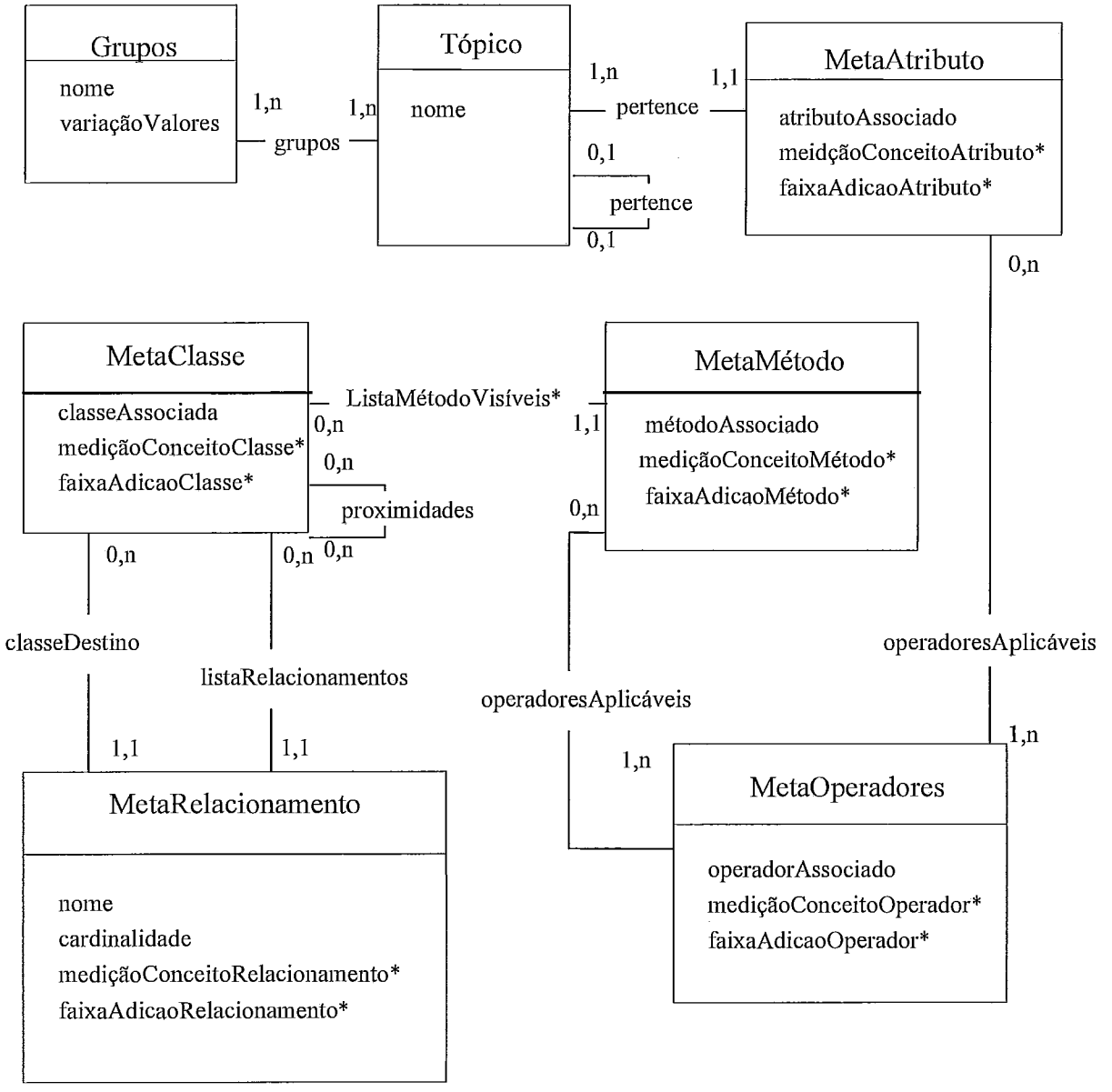


Figura 4.5 - Esquema da base de Metadados

Na Figura 4.5 apresentamos o esquema de dados da base de metadados. Podemos verificar que existem informações neste esquema de dados que serão utilizadas pelos dois módulos da camada de conhecimento:

- *pelo módulo OQLBot, para o relaxamento da consulta.*
- *pelo módulo IntBot, em conjunto com as informações provenientes do Tradutor de Conhecimento para construção e manutenção da interface do usuário. Os atributos utilizados para esta tarefa estão assinalados com o símbolo *.*

As classes "Tópico" e "Grupos" são utilizadas para realizar o relaxamento da consulta, na primeira forma definida pelo *OQLBot*. Os atributos são classificados em tópicos, sendo que atributos diferentes podem fazer parte de um mesmo tópico e os tópicos podem ser agrupados em outros tópicos. A classe "Grupos" apresenta o agrupamento de valores possíveis para os atributos, armazenando em "variaçãoValores" os possíveis valores de cada grupo do tópico relacionado.

As classes com prefixo "Meta" armazenam meta informações sobre os elementos de consulta associados. Os atributos com o sufixo "Associado(a)" armazenam qual o elemento do esquema de dados a que o objeto se refere. Os atributos com o prefixo "faixaAdição" são utilizados para a adaptação da interface durante uma sessão do usuário.

Os atributos com o prefixo "mediçãoConceito" são utilizados para classificar os elementos de consulta de acordo com os conceitos, criados para as regras de adaptação. Através deste atributo é que se realiza a pré-classificação dos elementos, permitindo ao módulo *IntBot* selecionar o grupo de elementos de consulta correspondente ao conceito determinado para apresentação pelo *Tradutor de Conhecimentos*.

Na classe "MetaClasse" o atributo proximidades é utilizado para realizar a adaptação da interface durante uma sessão já descritos no módulo *IntBot*.

Na classe "MetaRelacionamento" o atributo "nome" informa o nome do relacionamento na classe origem, o atributo "classeDestino" informa qual a classe objetivo deste relacionamento e o atributo "cardinalidade" como é a relação entre a classe origem e a classe destino do relacionamento, no que se refere a quantidade de elementos da classe destino. A cardinalidade pode ser do tipo 1:1 indicando a existência de um único objeto da classe destino ou 1:n indicando um conjunto de objetos da classe

destino. Essas informações são utilizadas para a geração da consulta declarativa pelo *OQLBot*.

4.4 Conclusão

A grande difusão de informações, através da utilização da rede mundial de computadores, aliada a maciça utilização de computadores pelos mais diversos usuários foi o grande fator impulsionador do estudo de sistemas de consulta capazes de atender melhor a usuários inexperientes.

Ao propor esta arquitetura, pretendemos solucionar algumas falhas desse rápido processo de difusão dos computadores e dos sistemas de bases de dados entre usuários não especialistas em informática, atacando principalmente o processo de construção de consultas.

As bases de regras, da forma como foram idealizadas nesta arquitetura, são independentes do esquema de dados que esta sendo consultado. Caso haja necessidade de definição de regras específicas para um esquema de dados, deve-se utilizar uma base de regras separada, para não restringir o sistema de consulta. A mesma idéia pode ser utilizada para a base de metadados.

Com relação aos critérios de classificação de sistemas de consulta apresentados no capítulo sobre Sistemas de Consultas Visuais, podemos dizer que a arquitetura proposta imprime aos sistemas construídos algumas características relacionadas a esta classificação.

A estratégia de interação utilizada na atividade de compreensão do domínio de interesse da consulta é a de refinamento de cima para baixo, já que o processo de adaptação da interface durante uma mesma sessão caracteriza-se, entre outras atividades, por aumentar o número de elementos apresentados ao usuário expandindo sua visão do esquema de dados da base consultada. Já na formulação da consulta, a arquitetura caracteriza os sistemas por utilizar subconsultas, compondo conceitos intermediários e podemos dizer também que algumas idéias do conceito de navegação na parte intencional do esquema são utilizadas, já que os relacionamentos entre as classes são explicitamente utilizados na criação da consulta.

O armazenamento e a reutilização de consultas é implantado nesta arquitetura, de forma que o usuário possa salvar consultas realizadas e alterar seus parâmetros durante

a nova execução e ainda utilizar uma consulta armazenada como parte de uma nova consulta construída.

A adaptação do sistema ao usuário é a característica mais marcante desta arquitetura e portanto será obtida nos sistemas a serem desenvolvidos. Ao contrário dos demais sistemas apresentados, as facilidades dos sistemas ocorrem no sentido sistema usuário e não no sentido inverso.

Capítulo 5 - Protótipo

A implementação de um sistema de consulta, baseado na arquitetura descrita no capítulo anterior, será apresentada neste capítulo. Esta implementação foi desenvolvida utilizando-se a linguagem JAVA™, como linguagem de programação e de criação das interfaces visuais, a máquina de inferência ESSLN(LE,V.T., 1990) e o Gerente de Objetos Armazenados(GOA++) (MAURO, 1998) desenvolvido na COPPE, para acesso a bases de dados orientada a objetos.

Este capítulo encontra-se dividido em quatro seções. A primeira seção é uma introdução à implementação realizada, ressaltando principalmente as características de cooperação descritas para a arquitetura que foram implementadas neste sistema. A segunda seção descreve de forma detalhada como foi feita a implementação do sistema, qual a arquitetura de implementação utilizada e a base de regras de adaptação. A terceira seção apresenta um exemplo de utilização do sistema que permite verificar as funcionalidades desenvolvidas. E finalmente na quarta seção apresentamos algumas conclusões sobre a implementação deste protótipo.

5.1 Introdução

Nem todas as características de cooperação, apresentadas no capítulo anterior para a arquitetura, foram implementadas neste protótipo, como mostra a Tabela 5.1 apresentada abaixo. Somente as características marcadas com *X* foram realmente implementadas.

O paradigma visual selecionado para a implementação deste protótipo foi o de ícones e a manipulação direta, que serão discutidos ainda nesta seção.

A adaptação da interface foi a característica tratada com maior enfoque nesta implementação, pela inexistência de sistemas de consulta, retratados na literatura, que

utilizem este paradigma para melhorar o processo de interação com seus usuários. Tanto a adaptação entre as sessões de um usuário, quanto durante uma mesma sessão foram realizadas como descrito no capítulo anterior.

Característica de Cooperação	*Implementada*
paradigmas visuais	*X*
adaptação da interface entre sessões e durante a sessão	*X*
visualização da consulta(OQL) gerada	*X*
possibilidade de armazenamento de consultas	*X*
utilização de consultas salvas como componentes de outras consultas	
possibilidade de visualizar subesquemas ou visões do modelo	*X*
criação automática dos relacionamentos entre entidades na consulta	*X*
apresentação dos objetos relacionados de um objeto no resultado da consulta	*X*
relaxamento de consultas	

Tabela 5.1 – Tabela de características cooperativas.

A adaptação durante a sessão do usuário é realizada somente em algumas sessões, de acordo com as regras de adaptação utilizadas no sistema. A característica mais interessante desta adaptação é que ela só ocorre quando o sistema infere que já houve uma "acomodação", no modelo mental do usuário, das alterações realizadas na última adaptação. Dessa forma, buscamos alterar a interface de forma amena, minimizando o sentimento de perda de controle por parte do usuário comum nas interfaces adaptativas, em virtude das constantes mudanças na interface. Outras formas de adaptação podem ser mais agressivas, determinando alterações na interface em cada nova sessão iniciada pelo usuário.

Ao criar a sentença de ícones que representa a consulta desejada pelo usuário é possível verificar como é a consulta declarativa gerada para àquela sentença. Além disso, pode-se armazená-la com um nome para posterior utilização. Neste protótipo não implementamos a possibilidade de alterar os parâmetros de uma consulta armazenada nem de utilizá-la como sub-componente de outra consulta.

A visualização de subesquemas ou visões do modelo é realizada de forma intrínseca pelo sistema, através do processo de adaptação da interface. Como o sistema seleciona quais os elementos de consulta que são visíveis ao usuário, implicitamente se determina quais as partes do modelo de dados consultado serão vistos pelo usuário.

Os relacionamentos entre as classes são criados automaticamente pelo sistema, durante o processo de construção da sentença de ícones(consulta). Como citado no capítulo anterior, a criação automática ocorre somente para os relacionamentos selecionados pelo sistema para o usuário.

Os resultados das consultas serão apresentados por este sistema através de tabelas de dados, mas quando o valor do atributo for um relacionamento, um botão será disponibilizado ao usuário permitindo a navegação entre os objetos relacionados.

O relaxamento de consultas não foi implementado neste sistema em virtude do tempo, e ao contrário da técnica de adaptação de interface, o relaxamento já foi implementado e testado em alguns sistemas de consulta como apresentam CHU *et. al.*(1993) e RAMOS *et. al.*(1996).

5.1.1 Paradigma Visual

Baseados nos estudos realizados por CATARCI *et al.*(1991), escolhemos como paradigma visual para esta implementação a manipulação direta e os ícones. Todos os elementos de consulta são apresentados na interface através de ícones e podem ser selecionados e arrastados na interface, e ainda, dependendo do elemento de consulta representado, é possível aplicar funções específicas.

Sobre o uso da manipulação direta na realização de consultas, EBERTS e BITTIANDA(1993) afirmam que apesar deste paradigma ser mais rápido na realização de tarefas complexas, sua utilização para realização de consultas é dificultada em virtude da complexidade do problema e do volume de conhecimento que o usuário necessita para realizar esta tarefa. Em nossa arquitetura, minimizaremos essa dificuldade, diminuindo os conhecimentos que o usuário necessita para realizar a tarefa.

5.1.1.1 Manipulação Direta

A manipulação direta representa a sensação do usuário em operar diretamente sobre os objetos apresentados, utilizando as técnicas de apontamento e seleção. Ao

invés de fornecer comandos na interface, que atuem sobre determinado objeto, o usuário atua sobre estes na interface, determinando a realização da operação desejada.

As principais vantagens em seu uso são a facilidade de aprendizado e memorização, os objetos vistos na tela são os objetos que o sistema possui(WYSWYG), a interface é flexível com ações reversíveis, o retorno das ações é visual(de acordo com o contexto) e permite uma redução do número de erros cometidos pelos usuários.

Entretanto, apresenta como desvantagens principais o fato de não ser auto-explicativa, podendo se tornar ineficiente, a necessidade de selecionar os ícones corretos, ou seja, que permitam ao usuário uma associação entre seu modelo mental e a representação selecionada e a difícil utilização das interfaces quando o número de objetos e/ou ações é muito grande(GENTNER e NIELSON, 1996).

5.1.1.2 Ícones

Os ícones são pequenas imagens utilizadas para representar objetos e ações, ou ainda, dados e processos, em programas de computador a fim de realizar a comunicação entre o usuário e a aplicação. São caracterizados por proporcionar às pessoas um reconhecimento instantâneo de símbolos familiares.

De acordo com CHANG(1988), os ícones utilizados nas linguagens visuais são compostos por duas partes distintas: uma lógica, que possui seu significado e uma física que é a imagem a ser apresentada. Os ícones devem ser desenvolvidos baseados em um contexto e nunca operam isoladamente, na realidade deve-se adotar um sistema de ícones(CHANG, 1988), que consiste em um conjunto estruturado de ícones relacionados, armazenados em um dicionário de ícones.

Os ícones são utilizados para representar diversos tipos de objetos e ações de uma aplicação. Os tipos de objetos a serem representados podem ser: objetos naturais encontrados na natureza, que são facilmente representados pela sua imagem real e já conhecida pelas pessoas; objetos artificiais, mas que existem fisicamente e por isso possuem uma forma já conhecida e podem ser representados por essa forma; e finalmente os objetos abstratos, os mais difíceis de representar pois não existem no mundo real e normalmente são expressos através de símbolos.

As ações também são difíceis de representar, pois possuem embutidas uma dinâmica, que não é passada para os ícones, que são estáticos. Dessa forma, BAECKER *et al.*(1991) defendem a animação dos ícones, a fim de, enriquecer sua capacidade de representação, facilitando o entendimento, tanto de usuários novatos quanto experientes. Outra forma de enriquecimento sugerida por NIELSEN(1993) é a associação de sons aos ícones.

Em alguns casos, é necessário utilizar a combinação de dois ou mais ícones conhecidos para representar um objeto ou ação, ou ainda acoplar ao ícone algum texto para facilitar sua compreensão.

Existem algumas estratégias defendidas por HUANG(1988) para facilitar o desenvolvimento e a aceitação de interfaces baseadas em ícones que foram utilizadas no desenvolvimento deste protótipo. Podemos citar entre essas estratégias:

- *dependência cultural e da aplicação*

  O ícone deve ser desenvolvido considerando-se sua semântica na cultura em que o sistema será inserido bem como de acordo com a aplicação a que se destina.

- *fácil reconhecimento*

  Um ícone deve ser reconhecido e memorizado por suas formas, que devem apresentar um significado preciso.

- *distinção de outros ícones do sistema*

  Todos os ícones de um mesmo sistema devem possuir alguma peculiaridade que o diferencie dos demais ícones do sistema.

- *consistência no ambiente*

  O desenho e o formato dos ícones devem ser consistentes dentro de um mesmo ambiente.

5.2 O Sistema de Consulta

O sistema de consulta implementado é baseado em uma arquitetura cliente servidor, onde o cliente pode ser executado nos diversos locais da rede mundial de computadores e o servidor é executado em um único local, que tem acesso aos dados armazenados, tanto os de conhecimento quanto os da base de dados consultada.

Outro aspecto importante que será considerado nesta seção é a forma de apresentação selecionada para o sistema, ou seja, como são as interfaces de interação com o usuário construídas neste prótotipo.

5.2.1 Arquitetura do sistema

Como já dissemos, a arquitetura utilizada para implementar o sistema de consulta, apresentada na Figura 5.1 abaixo, é baseada na técnica cliente servidor. Desta forma, é possível existir diversos clientes e um único servidor atendendo a todos.

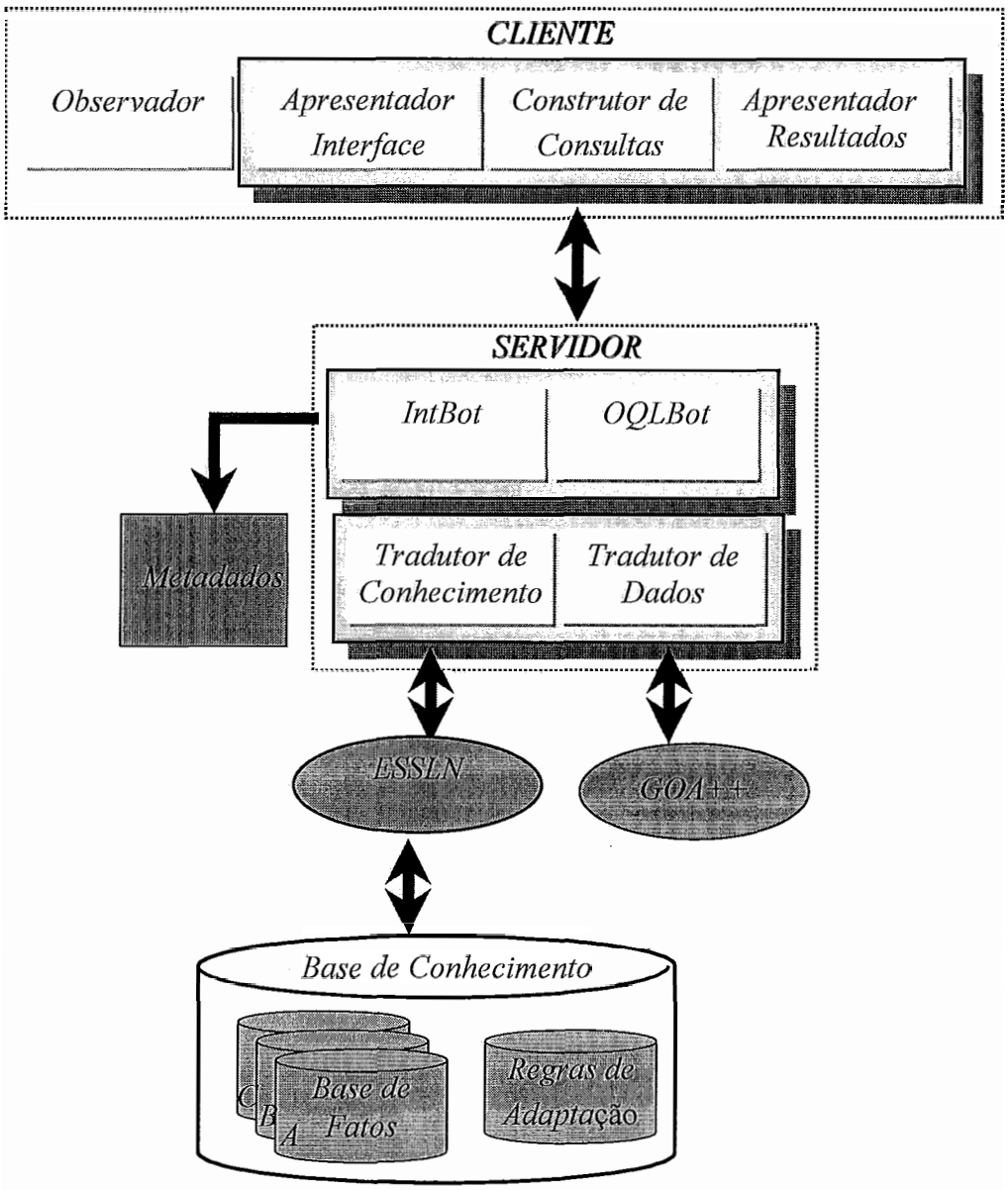


Figura 5.1 – Arquitetura da implementação do sistema de consulta.

Tanto o cliente quanto o servidor desta aplicação foram implementados em linguagem JAVA™. No cliente foi implementada a Camada de Interface, enquanto no

servidor foram implementadas a Camada de Conhecimento e a Camada de Tradutores, descritas na arquitetura proposta nesta tese. O cliente e o servidor se comunicam através de troca de mensagens utilizando *sockets*.

O módulo Observador foi implementado através de métodos de uma classe escrita em JAVA™ que implementa a janela de consulta do sistema. Estes métodos informam ao servidor todas as ações do usuário na interface.

O módulo Apresentador de Interface foi também implementado através de métodos da janela de consulta do sistema e é responsável por receber do servidor mensagens determinando a apresentação de elementos de interface e de elementos de consulta. Estes métodos tratam essas mensagens e determinam como e onde os elementos serão apresentados.

O módulo Apresentador de Resultados possui uma classe específica que recebe os dados a serem apresentados e constrói a tabela de apresentação. As mensagens que informam os dados contém tanto dados de tipo atômico quanto relacionamentos e este módulo separa estes elementos e os apresenta de forma diferente na tabela(texto e botão).

O módulo Construtor de Consultas foi implementado em uma classe específica chamada Painel de Desenho. Esta classe gerencia todo o processo de construção das sentenças de ícones e todos as funcionalidades disponibilizadas neste processo como inclusão e remoção de ícones na consulta, movimentação de ícones na área da consulta, adição de relacionamentos e outras. Este módulo também envia ao servidor a consulta construída para que este informe os resultados obtidos.

O módulo IntBot é responsável por todo o processo de adaptação da interface. Este módulo constrói e envia as inferências para o módulo Tradutor de Conhecimento, que as envia para a máquina de inferência e obtém as informações que determinam que elementos de interface e de consulta devem ser apresentados.

O módulo OQLBot ficou restrito neste protótipo por não ter sido implementado o processo de relaxamento de consultas. Este módulo foi implementado através de métodos que armazenam as informações provenientes das consultas realizadas pelos usuários, além de construir a consulta em linguagem declarativa e em seguida determinando sua execução através do Tradutor de Dados.

A máquina de inferência ESSLN(LE,1990) é uma *shell*, que permite escrever as regras e fatos em uma linguagem intermediária entre a linguagem PROLOG e o português. Essa *shell* roda sobre o SWI-PROLOG(WIELEMAKER,1997), uma implementação do PROLOG, que pode ser executada em micro computadores em ambiente WINDOWS. O objetivo da utilização desta *shell* é facilitar o processo de construção das regras de adaptação e dos fatos do modelo do usuário .

O Tradutor de Conhecimento encontra-se implementado parte em uma classe escrita em JAVA™ e outra parte em uma biblioteca, implementada em linguagem C. Esta biblioteca garante o acesso à máquina de inferência, escrita em SWI-Prolog, a partir das classes escritas em JAVA™, sendo capaz de enviar as inferências provenientes do servidor para a máquina e no sentido contrário, enviar as respostas produzidas pela máquina ao servidor.

O Tradutor de Dados é composto por uma classe implementada em JAVA™ e um pacote de classes chamado de Servidor GOA++, desenvolvida por MAURO(1998). Este pacote realiza a troca de mensagens entre o servidor do sistema de consulta e o GOA++, garantindo com isso a execução das consultas e apresentação dos resultados.

É importante ressaltar neste contexto que o servidor da aplicação roda em ambiente Windows 95 e o Gerente de Objetos Armazenados(GOA++) que atua como o SGBDOO roda em ambiente UNIX. Toda a comunicação é transparente ao usuário e realizada pelo módulo Tradutor de Dados.

A base de conhecimento do protótipo utiliza duas bases: a base de regras de adaptação e a base de fatos do usuário, representada na arquitetura proposta no capítulo anterior pelo modelo do usuário. A primeira base é exatamente como foi descrita e será detalhada na próxima seção. A segunda base, base de fatos do usuário, é gerada a cada novo acesso do usuário ao sistema, a partir das informações contidas em um histórico de utilização do sistema pelo usuário.

Desta forma, durante uma sessão do usuário, o seu histórico de utilização é atualizado, a partir de suas interações com o sistema e a base de fatos se mantém estável. Na próxima sessão do usuário, uma nova base de fatos será gerada, baseada no novo histórico do usuário obtido com seu último acesso. Esta forma de trabalho foi adotada para facilitar o processo de manutenção das informações de conhecimento do

usuário no histórico do usuário, sem ficar restrito ao que seria compreendido pela máquina de inferência.

Para maiores detalhes de implementação, o Apêndice II apresenta os modelos de projeto do cliente e do servidor.

5.2.2 Interfaces

O sistema utiliza três interfaces básicas: interface de seleção do banco de dados, interface para realização da consulta e a interface para apresentação dos resultados.

A primeira interface é simples e baseia-se na apresentação de uma lista de ícones que representam as bases de dados disponíveis para consulta. A interface de consulta é um pouco mais complexa e baseia-se na estrutura apresentada na Figura 5.2.

A área de apresentação de consultas salvas apresenta a lista de consultas armazenadas do usuário. Esta área pode ou não existir na interface de um usuário, sendo determinado pelo servidor na construção da interface.

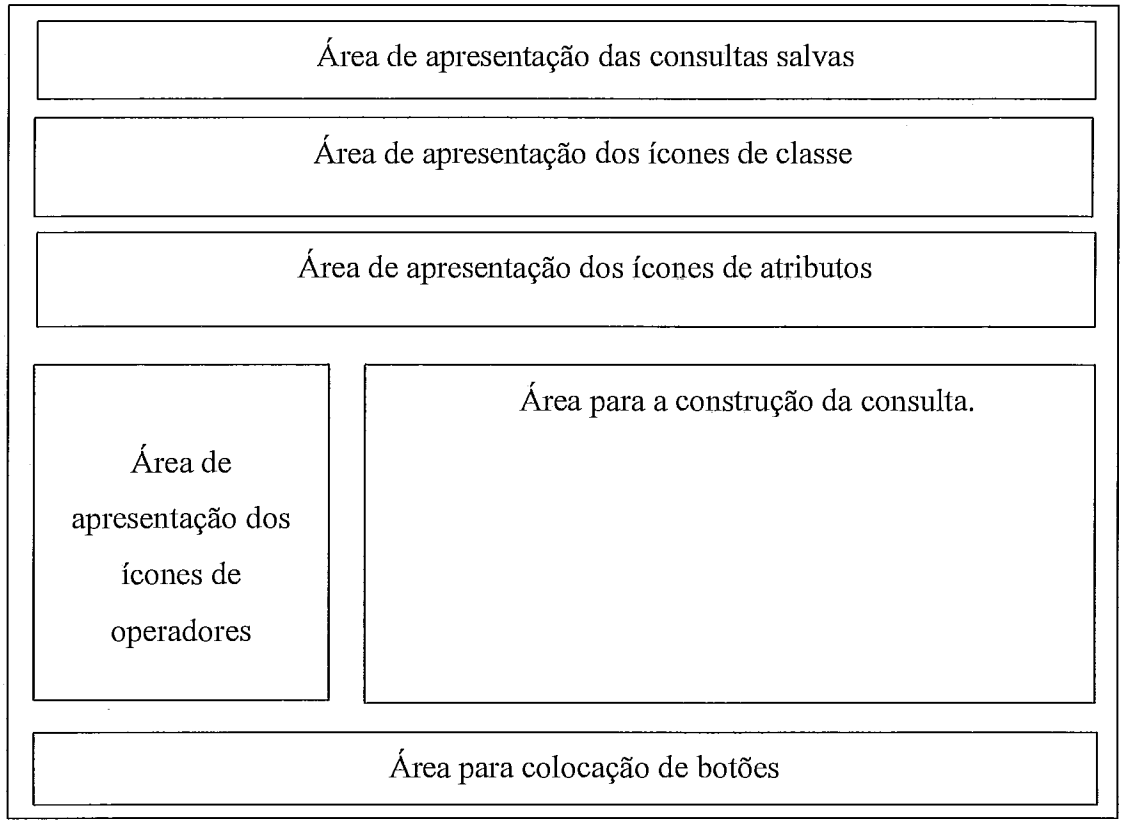


Figura 5.2 – Interface básica para criação de consultas.

A área de apresentação dos ícones de classe apresenta as classes selecionadas para a interface do usuário, utilizando o paradigma visual de ícones. Da mesma forma,

se constrói a área de apresentação dos ícones de atributos e de operadores. Em todas essas áreas é possível adicionar e retirar os elementos de consulta correspondentes, caracterizando uma adaptação da interface dirigida pelo usuário.

A área para construção de consultas é destinada à construção das sentenças de ícones, que se transformarão em consultas, através do processo de arraste e solte dos elementos das áreas de apresentação(classes, atributos e operadores).

A sentença de ícones é construída de acordo com as regras do módulo Construtor de Consultas, apresentadas no capítulo anterior. Em nossa implementação, adicionamos algumas características às sentença de ícones buscando facilitar o processo de aprendizagem do usuário sobre o modelo de dados consultado, explicitando os tipos de relações entre os elementos de consulta e a aplicação dos operadores.

Para explicitar a relação entre os elementos de consulta, definimos que os atributos e os relacionamentos trariam consigo uma conexão associada, diferenciada de acordo com a relação estabelecida. Assim, os atributos são ligados às suas classes através de uma linha verde, enquanto os relacionamentos são representados por linhas brancas orientadas e nomeadas, onde a orientação indica a classe destino do relacionamento. A aplicação dos operadores é ressaltada ao usuário através do envolvimento dos elementos que participam da operação por linhas tracejadas da cor vermelha. Essas novas características permitem a construção das sentenças seguindo um formato em árvore.

A última área, chamada área para colocação de botões, é utilizada para apresentar botões que executam funções gerais do sistema. Os botões utilizados neste protótipo executam as seguintes funções: salvar consultas, executar consultas, pedir auxílio, cancelar uma consulta e sair da janela de consultas. Esses botões são fixos e sempre ocorrem para todos os usuários. Para os usuários que não possuem área de apresentação de consultas salvas, um outro botão, que permite visualizar e selecionar as consultas armazenadas pelo usuário, é disponibilizado nesta área.

A interface de apresentação de resultados é simples, baseada na apresentação dos valores em forma de tabelas, com a combinação de colunas de tipos simples e colunas de botões para acessar os relacionamentos.

5.2.3 Regras de adaptação

A forma como as regras de adaptação são construídas, tanto no nível sintático quanto semântico, consiste em uma decisão de implementação do sistema e não faz parte da arquitetura proposta.

No nível sintático, as regras em nosso protótipo foram escritas em uma linguagem intermediária entre o português e o PROLOG, interpretadas pelo ESSLN*.

No nível semântico as regras forma definidas utilizando as informações sobre o modelo de usuários, descritas no capítulo anterior, que são mantidas e atualizadas pelo sistema. Nesta seção, iremos apresentar as regras de adaptação utilizadas, de uma forma gráfica.

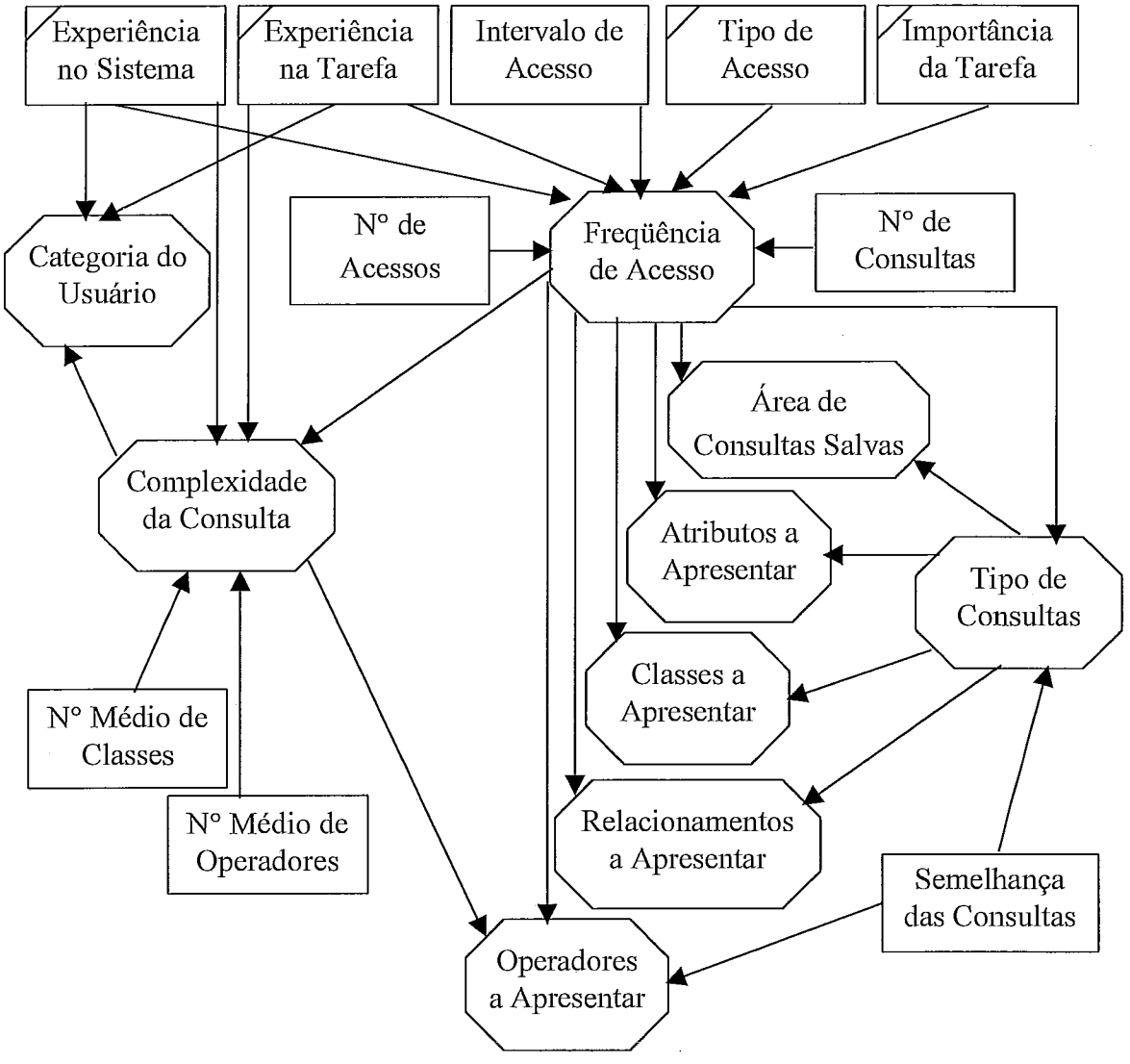


Figura 5.3 – Organização dos fatos e regras para construção da interface adaptativa.

Na Figura 5.3 apresentamos os relacionamentos de dependência entre os diversos fatos e regras utilizados para a construção da interface entre sessões. Os elementos gráficos representados pelo retângulo são os fatos e os hexágonos são as regras. As setas indicam a dependência entre esses elementos, sendo o elemento da ponta da conexão orientada o que depende do elemento na origem da conexão. Os retângulos cortados representam os fatos que não podem ser determinados pelo sistema e precisam ser informados pelo usuário.

Podemos verificar pela Figura 5.3 a existência de quatro fatos que precisam ser informados pelo usuário. É claro que perguntar simplesmente ao usuário, por exemplo, se ele tem experiência na tarefa ou no sistema pode acarretar em respostas incorretas para o sistema. Assim, em nosso protótipo, optamos por associar à cada fato que precisa ser obtido do usuário um conjunto de perguntas, com pesos associados, que realmente possam informar ao sistema o fato sobre o usuário. Essas perguntas são configuráveis permitindo com isso incluir e retirar perguntas e ainda modificar os pesos associados.

Assim, por exemplo, para saber se o usuário tem experiência no sistema, foram elaboradas as perguntas, apresentadas na Tabela 5.2, sobre todas as funcionalidades disponibilizadas pelo sistema. De acordo com suas respostas, é possível saber se o usuário realmente tem ou não experiência no sistema. As perguntas elaboradas são bem específicas, para que respostas do tipo *sim* e *não* sejam suficientes, e também são claras para que o usuário não tenha dificuldades em compreendê-las.

Pergunta	*Peso*
Você foi treinado para utilizar este sistema ?	*15*
Você tem confiança nos conhecimentos que lhe foram passados ?	*25*
Você é capaz de realizar consultas corretas utilizando este sistema sem auxílio ?	*10*
Você sabe o que fazer quando um assunto(classe, atributo ou operador) não está disponível na janela de consultas ?	*25*
Você sabe como armazenar uma consulta ?	*20*
Você sabe como reutilizar uma consulta já existente ?	*25*

Tabela 5.2 – Perguntas sobre o fato experiência no sistema.

Entre esses fatos obtidos por perguntas ao sistema, existem dois que podem ser alterados com a utilização do sistema, que são a experiência no sistema e a experiência na tarefa. Para garantir a reciprocidade entre esses fatos e o conhecimento real do usuário, regras foram definidas para que a cada sessão do usuário, esses fatos possam ser reavaliados. Essas regras são baseadas no histórico de utilização do sistema pelo usuário.

As regras necessárias para a adaptação da interface durante uma mesma sessão são apresentadas na Figura 5.4. A separação foi feita somente para facilitar a visualização das dependências, pois na realidade compõem uma única base de regras de adaptação.

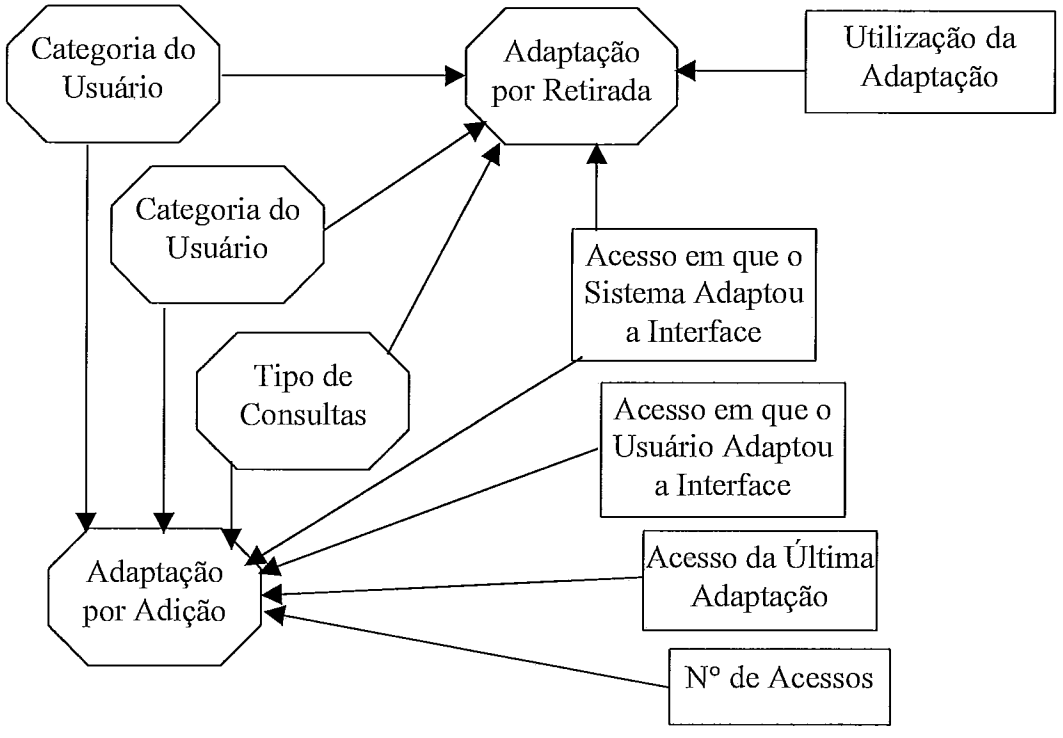


Figura 5.4 - Organização dos fatos e regras para adaptação da interface de consulta.

Durante a descrição da arquitetura no capítulo anterior, definimos a necessidade de se criar conceitos que agrupassem os elementos de consulta para facilitar o processo de inferência. Desta forma, neste protótipo foram criados os seguintes conceitos:

Para adaptação entre sessões de um usuário, os elementos de consulta foram agrupados em:

- *mais importantes*

  Apresentar os elementos mais importantes de acordo com o grau de importância definido por um profissional e armazenado na base de metadados.

- *mais usados*

  Apresentar os elementos mais usados de acordo a utilização dos elementos pelo usuário.

- *todos*

  Este conceito determina a apresentação de todos os elementos de consulta de uma determinada categoria.

Para adaptação durante uma mesma sessão do usuário, os elementos de consulta foram categorizados da seguinte forma:

- *mais próximos*

  Adicionar ou retirar os elementos mais próximos da classe mais usada, de acordo com a lista de proximidade para as classes definida na base de metadados.

- *por faixa*

  Adicionar ou retirar os elementos de acordo com a faixa de adição, definida na base de metadados para os elementos de consulta.

Foram definidas por volta de 110 regras para determinação da adaptação da interface que podem ser vistas com detalhes no Apêndice I.

5.3 *Utilização do sistema*

Nesta sessão vamos apresentar as interfaces que são construídas para um determinado usuário, cujo *login name* é *marieta.* O acompanhamento deste exemplo permitirá compreender como ocorre a adaptação da interface e como é o processo de construção de consultas neste sistema.

Inicialmente, o sistema apresenta ao usuário uma janela com a lista de bases de dados disponíveis para consulta, como mostra a Figura 5.5. Nesta tela, o usuário digita sua conta(*login)* e senha, e seleciona entre as bases de dados apresentadas na lista

horizontal de ícones, aquela que deseja consultar. No exemplo apresentado, selecionamos a base de ***Viagens***.

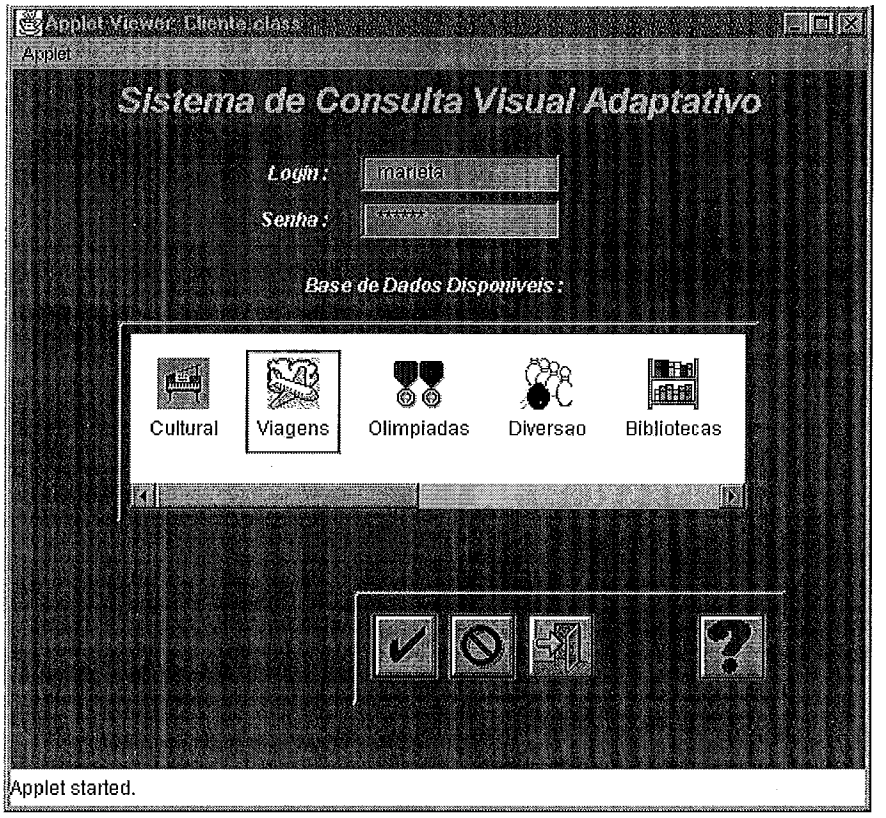


Figura 5.5 – Tela inicial do sistema.

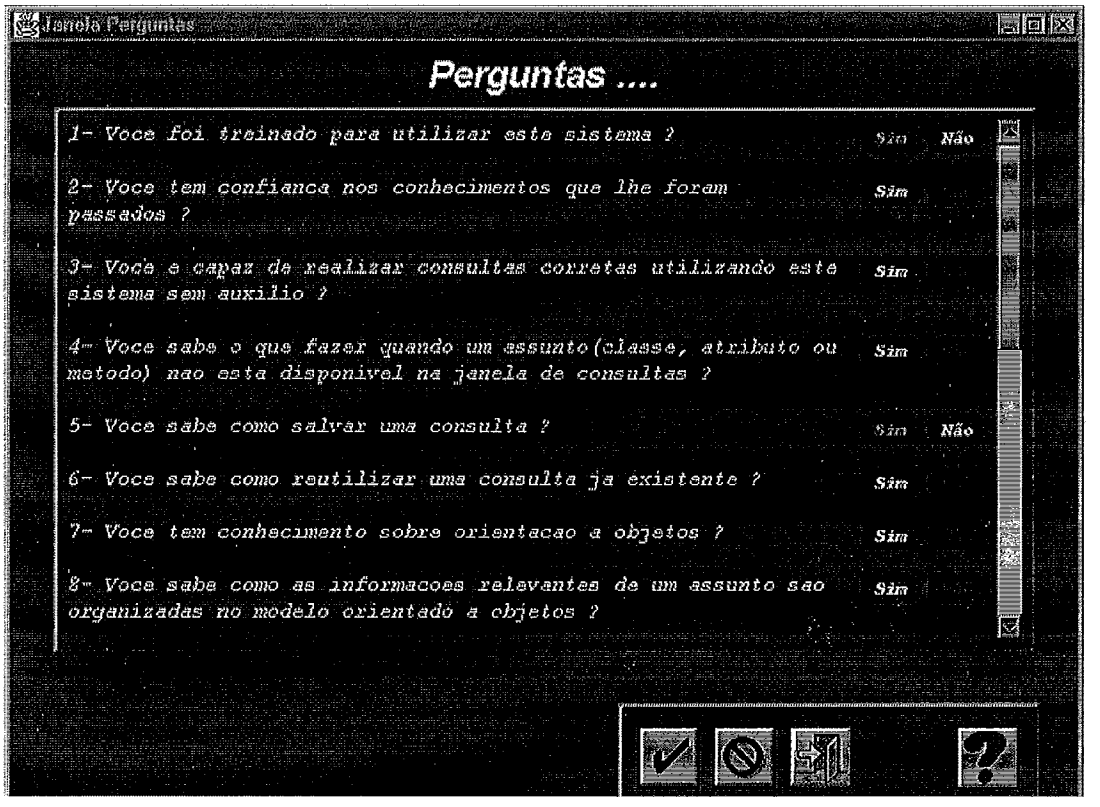


Figura 5.6 - Janela de perguntas do usuário.

Caso seja o primeiro acesso do usuário ao sistema, ou sua primeira consulta à base de dados selecionada, o sistema apresentará a Janela de Perguntas, que contém um conjunto de perguntas, do tipo *sim/não*, para estabelecer os fatos que não podem ser diretamente determinados pelo sistema, que são: experiência na tarefa, experiência no sistema, importância da tarefa e o tipo de acesso so usuário, se é por obrigação ou por opção. Através de suas respostas, o sistema determinará, baseado nos pesos associados de cada resposta, os fatos iniciais do sistema. Essa janela de perguntas é apresentada na

Figura 5.6.

Após o processamento das respostas do usuário, o sistema irá apresentar a Janela de Consultas, como mostra a Figura 5.7.

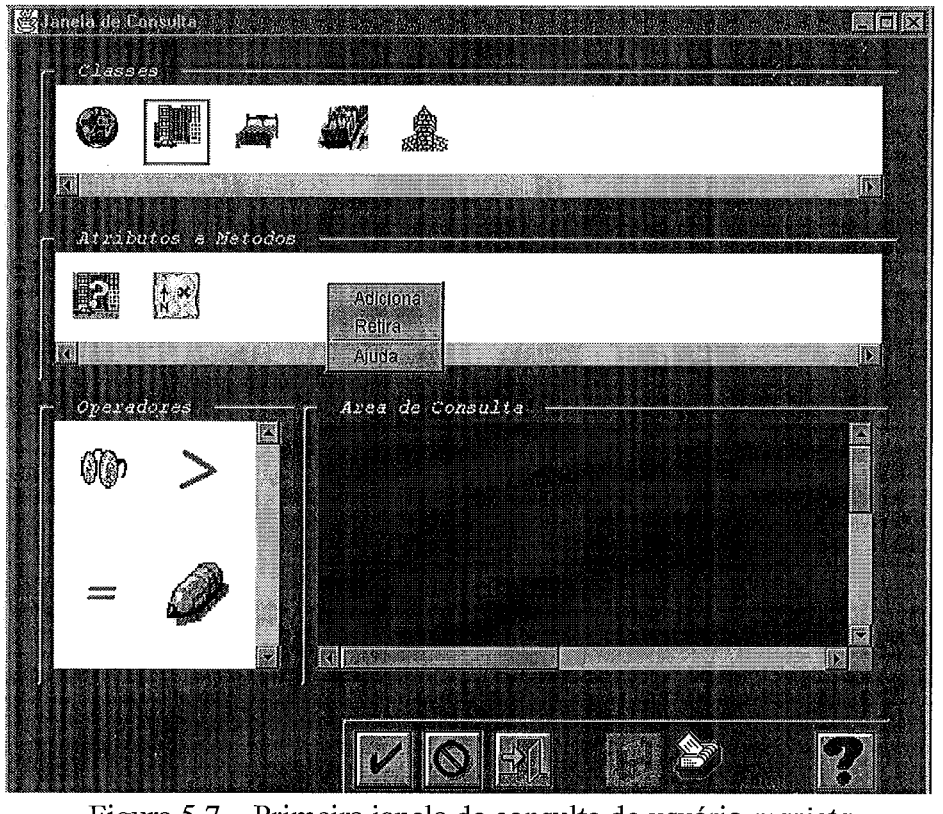


Figura 5.7 – Primeira janela de consulta do usuário *marieta.*

De acordo com as respostas fornecidas pelo usuário, o sistema decidiu apresentar os elementos de consulta mais importantes a este usuário. Assim, de acordo com as informações fornecidas na base de metadados, as classes, da direita para a esquerda, *Roteiro de Viagens, Cidade, Hotéis, Parques e Monumentos,* respectivamente, são apresentadas na área de ícones de classe.

Os atributos mais importantes incluídas na área de apresentação de ícones de operadores para a classe *Cidade* foram o *Nome* e a *Localização.* Os operadores incluídos, de cima para baixo e da direita para a esquerda, foram: operador para realização de projeção, chamado *Mostra*, operador de comparação *Maior que,* operador de comparação *Igual* e o operador que permite atribuir valores constantes, chamado de *Valor.*

Os botões incluídos na barra inferior da janela são da direita para esquerda: botão de execução da consulta, botão de cancela, para limpar a área de construção de consultas, botão de sair para finalizar a criação de consultas, botão de armazenamento de consultas, que no momento se apresenta não habilitado, sendo habilitado somente quando uma consulta for corretamente executada, botão que lista as consultas armazenadas e o botão de ajuda, ainda não implementado.

Em cada uma das áreas de ícones, o usuário pode selecionar com o botão direito do *mouse* um menu sensível ao contexto, que lhe permite adicionar novos elementos, retirar elementos e ainda requisitar auxílio, todas essas funções são sensíveis ao contexto onde o menu é disparado.

Esta funcionalidade pode ser verificada na Figura 5.7, onde o menu foi disparado da área de ícones de atributos e métodos. Se a opção selecionada for adicionar um elemento, uma lista com os atributos não visíveis será apresentada ao usuário e aquele que for selecionado será inserido na área de ícones de atributos. Se a opção desejada for a retirada de elementos, o elemento selecionado será retirado da área em questão. A opção de auxílio ainda não está implementada. Vamos supor em nosso exemplo que o usuário adicione o atributo *população* da classe cidade.

Para exemplificar a construção de uma consulta no sistema desenvolvido, suponha que desejamos saber *quais os roteiros de viagens que passam por cidades com mais de 500.000 habitantes ?*

Como já citamos, a consulta é construída arrastando-se a representação visual dos elementos de consulta para a área de construção de consultas, construindo uma seqüência visual de ícones. Assim, vamos supor que as classes *Roteiro de Viagens* e *Cidade* tenham sido arrastadas da área de apresentação dos ícones de classes para a área de construção de consultas, como mostra a Figura 5.8.

Podemos verificar que as classes se mantiveram disjuntas, ou seja, nenhuma conexão foi estabelecida entre elas, pois o relacionamento correspondente não foi

selecionado para a interface deste usuário. Dessa forma, o usuário deve estabelecer o relacionamento de forma manual, através do uso do menu sensível pela opção *Adiciona*. Assim, o sistema irá fornecer uma lista de relacionamentos originados na classe selecionada, para que o usuário selecione aquele que deseja, no caso o nome do relacionamento é *Passa por(Roteiro de Viagem Passa por Cidade)*.

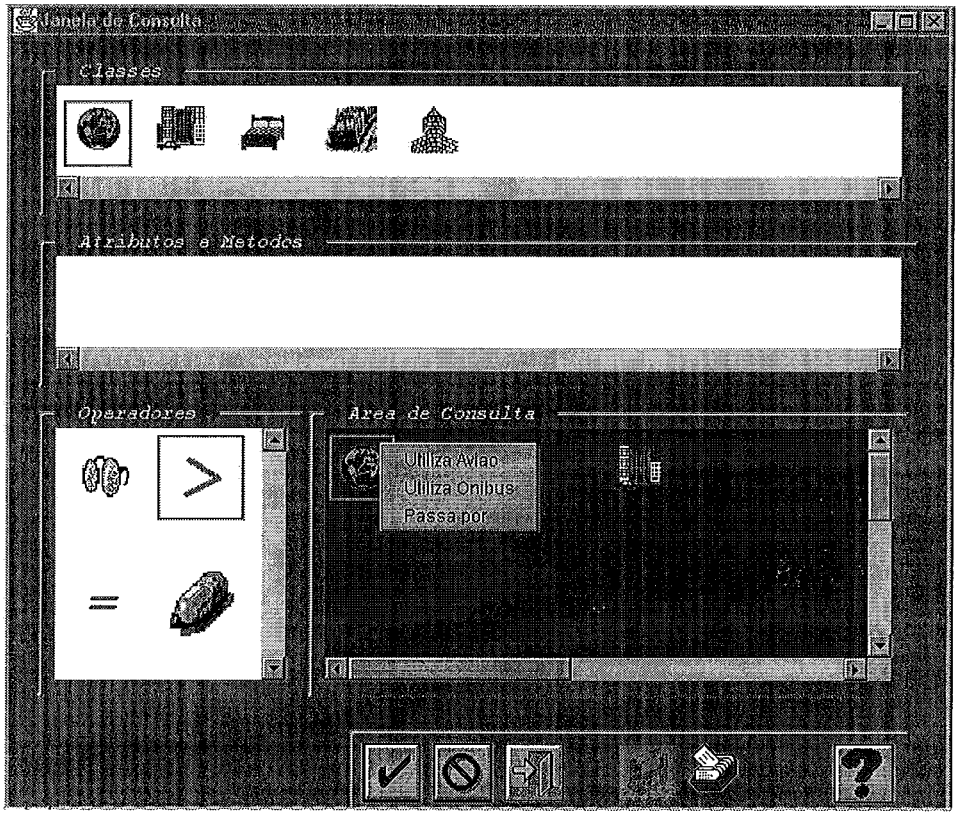


Figura 5.8 – Estabelecimento do relacionamento entre as classes da consulta.

O próximo passo é estabelecer o predicado da consulta, ou seja a restrição *população > 500.000*, como mostra a Figura 5.9. Para isso, o usuário deve arrastar para a seqüência de consultas o atributo *população*, que ele próprio inseriu na interface. Ao incluir este atributo na área de consultas, uma conexão se estabelece entre o atributo e sua classe. A seguir, será criado o predicado, arrastando-se para o lado direito do atributo *população* o operador *Maior que* e em seguida o operador *Valor*, onde através de um duplo clique uma Janela de Valor Associado será apresentada para que o usuário entre com o valor constante a ser utilizado no predicado. Automaticamente, quando o operador de comparação foi inserido à consulta, uma área retangular tracejada é criada, indicando uma seqüência visual que contém operadores. Para a consulta exemplo, a seqüência visual de ícones esta completa e portanto a consulta pode ser submetida ao gerenciador de objetos GOA++.

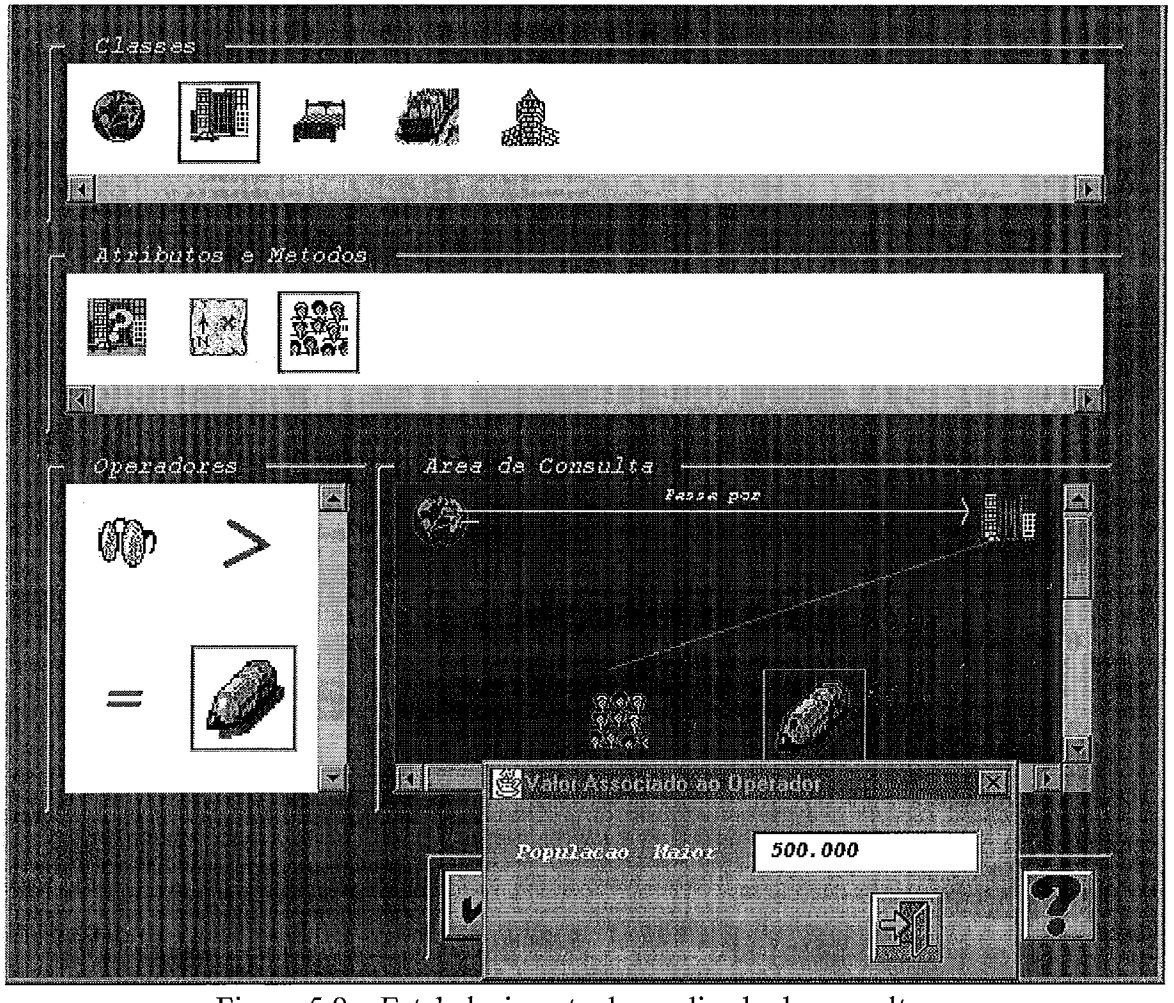


Figura 5.9 – Estabelecimento do predicado da consulta.

Os resultados desta consulta são apresentados na Figura 5.10. Para cada atributo dos objetos recuperados é criada uma coluna, cujo título é o próprio nome do atributo. Quando os atributos correspondem a relacionamentos, sua apresentação é realizada sob a forma de botões, permitindo que o usuário navegue pelos objetos relacionados ao objeto apresentado.

Para diferenciar quando uma janela de objetos corresponde ao resultado da consulta de quando foi obtida por navegação, decidimos utilizar títulos diferentes. Para as janelas de resultado da consulta, o título é "Resultado da Consulta", para as janelas de navegação o título é formado por "Navegação " mais o nome do relacionamento que está sendo navegado.

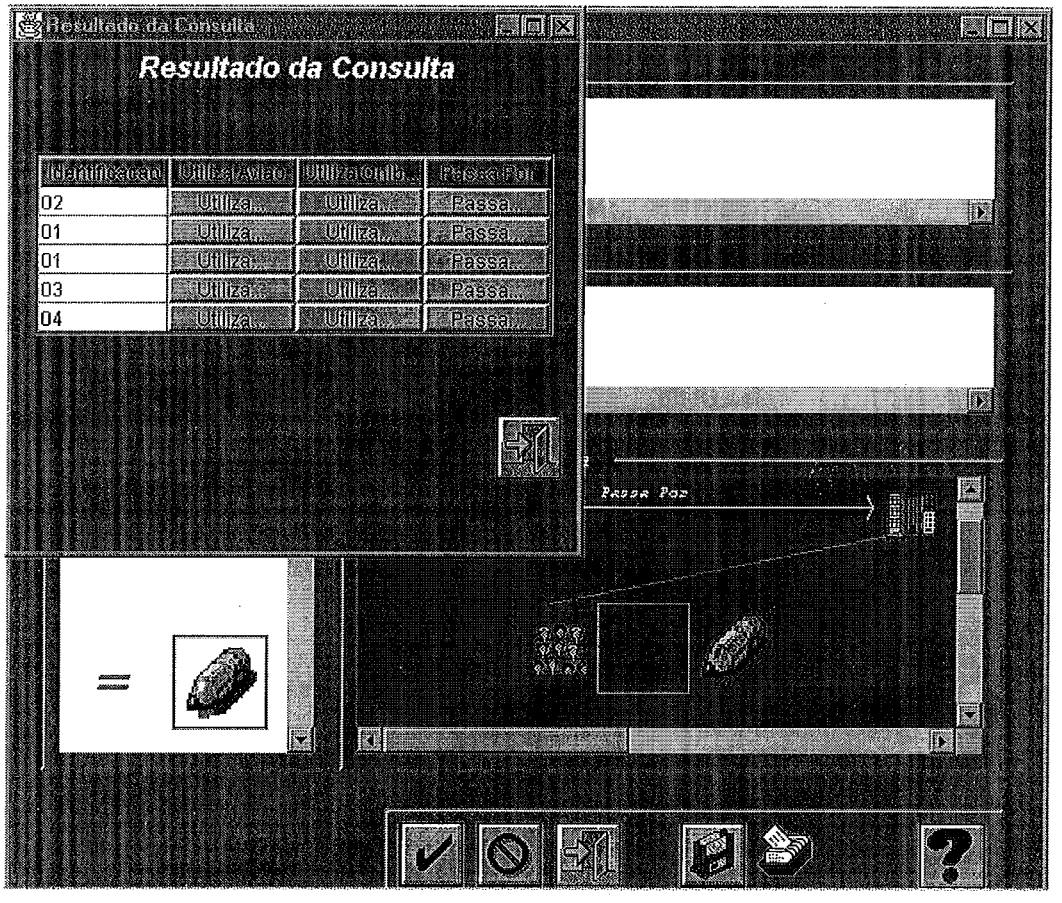


Figura 5.10 - Resultado da Consulta.

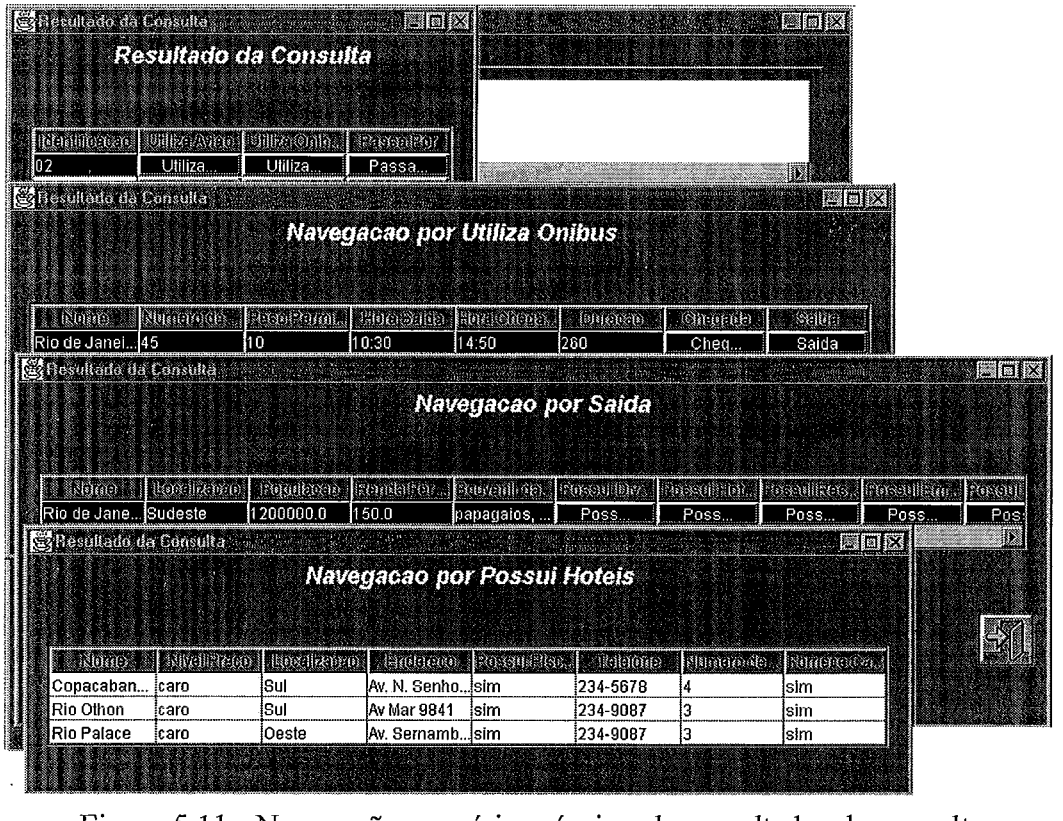


Figura 5.11 - Navegação em vários níveis pelos resultados da consulta.

Podemos verificar na Figura 5.11 uma navegação em vários níveis, realizada a partir do resultado da consulta anterior. O relacionamento escolhido para navegação foi *Utiliza Onibus*, do primeiro objeto da lista de resultados. Ao selecionar este relacionamento, uma nova janela de resultados é apresentada, com a lista de objetos do tipo ônibus vinculado ao objeto selecionado. Nesta lista, novas navegações são possíveis e assim sucessivamente sobre todos os relacionamentos apresentados.

Após um determinado tempo de uso do sistema pelo usuário *marieta,* o sistema realiza uma adaptação durante uma sessão do usuário, baseada na adição das classes mais próximas à classe *Cidade*. Dessa forma, foram adicionadas a interface do usuário as classes *Montanhas, Praias, Lagos e Museus* que correspondem a primeira zona de proximidade definida na base de metadados. A interface deste usuário se apresentará como mostra Figura 5.12.

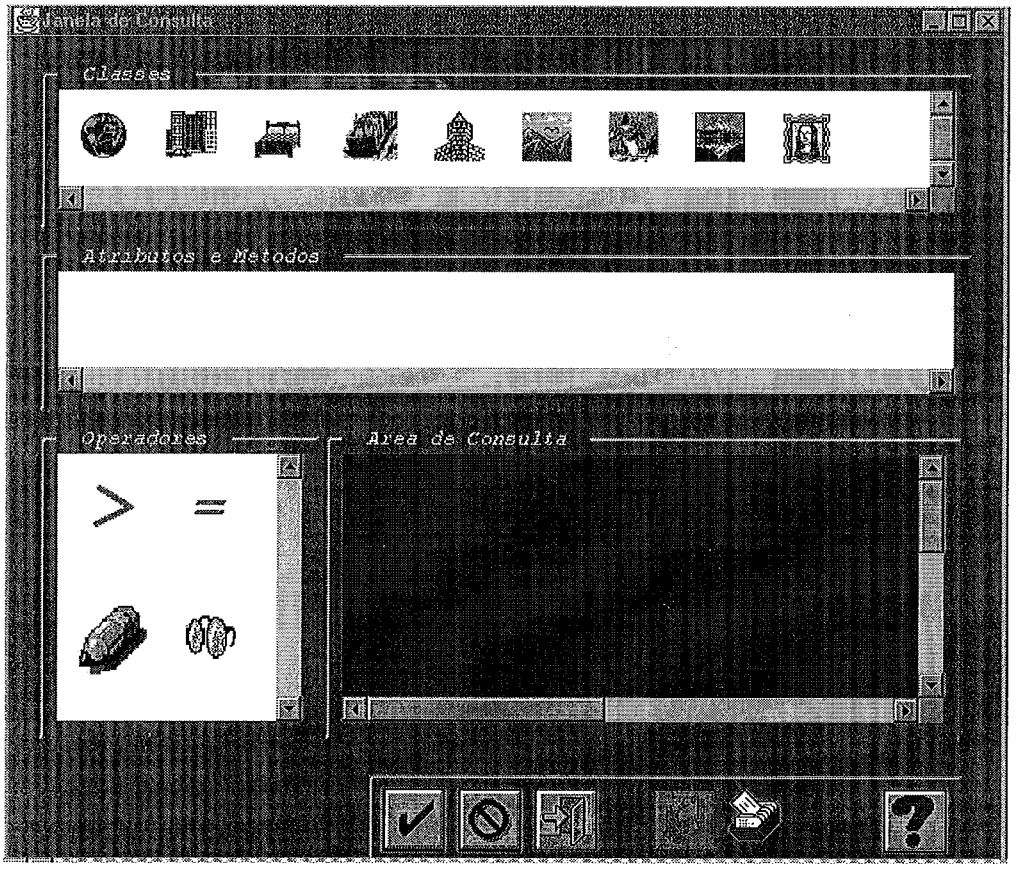


Figura 5.12 – Interface de consulta após a adaptação.

Para este mesmo usuário, uma nova adaptação entre sessões é realizada após outro intervalo de tempo, onde o usuário realiza várias consultas, produzindo a interface de consulta apresentada na Figura 5.13. Nesta interface, área de consultas

salvas foi preenchida com a lista de consultas armazenada pelo usuário. O botão(+) permite adicionar a consulta selecionada na lista à área de construção de consultas.

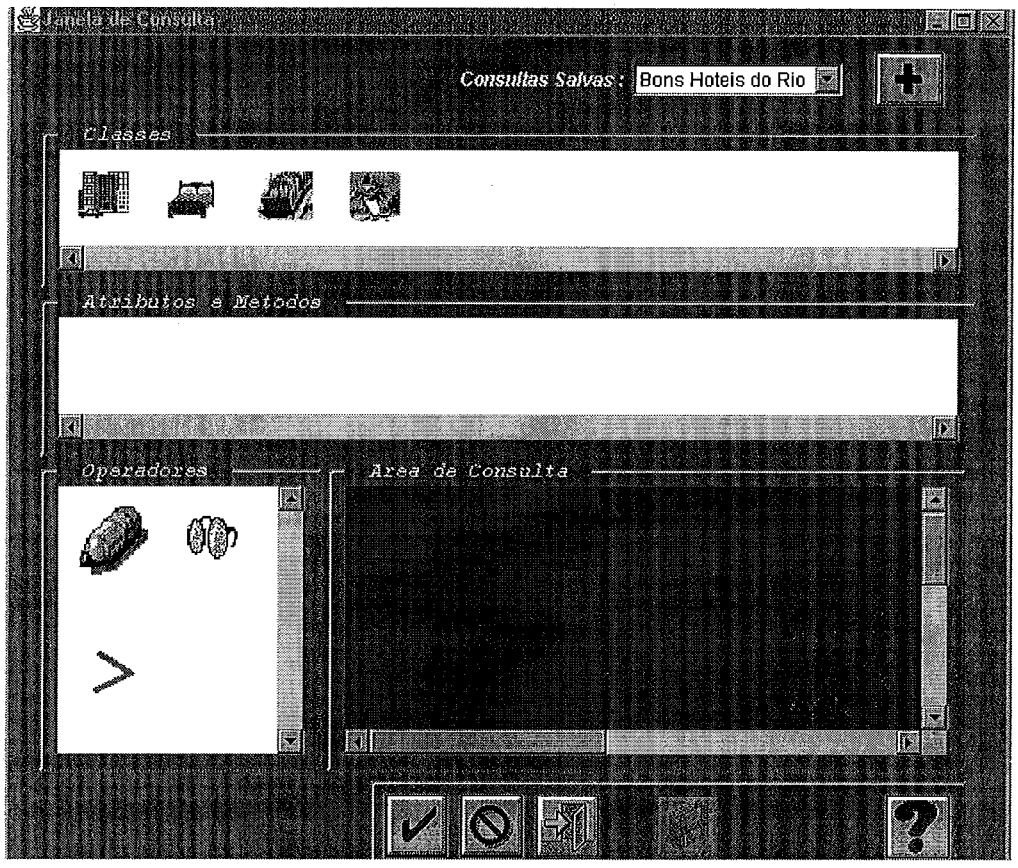


Figura 5.13 – Interface de consulta após adaptação.

Além disso, as classes apresentadas nesta interface são as mais utilizadas pelo usuário, *Cidade, Hotéis, Parques de Diversão e Praias*, respectivamente, bem como os operadores apresentados também são os mais utilizados.

5.4 Conclusão

Ao selecionar a linguagem JAVA™ como linguagem de programação do sistema de consulta, possibilitamos sua utilização na *internet*.

A escolha do SWI-PROLOG ocorreu principalmente por sua execução em ambientes Windows 95, o que facilitou bastante o desenvolvimento. Entretanto, seu desempenho em conjunto com a *shell* ESSLN foi muito baixo, tornando o sistema bastante lento nos momentos em que um número de regras relativas a um assunto é grande.

Este problema de desempenho não teria ocorrido se durante o processo de implementação do protótipo, a *shell* JESS(*Java Expert System Shell*) já estivesse

disponível para ser utilizada neste sistema, pois foi completamente desenvolvida em JAVA™ e evitaria a necessidade de criação de uma biblioteca em C e a consequente troca de contextos e formatos de representação dos dados que ocorreu para permitir a comunicação em JAVA™, C e SWI-PROLOG nos dois sentidos. Nosso protótipo realiza as adaptação durante uma mesma sessão do usuário basicamente no conteúdo da área de apresentação de ícones de classes, mas é possível realizar nas outras áreas também, a partir do conceito de faixas de adição.

Existe na literatura uma grande discussão retratada em (HUANG,1988, CAROLIS *et al.*, 1995, WATERWORTH *et al.*, 1993, FAMILANT e DETWEILER, 1993) sobre a universalidade da linguagem dos ícones e se realmente eles facilitam a comunicação entre o usuário e o sistema.

Apesar das dificuldades na escolha dos ícones, que requer a consideração de fatores como a cultura dos usuários, seu nível educacional e o ambiente em que vivem, ainda assim é um consenso entre os autores de que eles propiciam um aprendizado e uma memorização mais eficiente do que as outras formas de interação, aumentando significativamente o desempenho na realização das tarefas.

Alguns autores são mais profundos em sua análise sobre a correta escolha dos ícones, como CAROLIS *et al.*(1995) e defendem a adaptação dos ícones de acordo com as necessidades e características dos usuários.

Em nossa implementação cumprimos todas as fases do processo de adaptação descrito por BROWNE *et al.*(1990). A primeira fase é a realização do projeto da interface, avaliando quais as possibilidades existentes. A segunda fase é a escolha da variação de interface a ser utilizada. A terceira fase é a realização da avaliação, verificando se as adaptações promovidas foram úteis para o usuário, que é feita implicitamente pelo sistema, através do acompanhamento de uso das classes adicionadas pelo sistema.

Capítulo 6 - Conclusão

Os sistemas de consulta são responsáveis pelo principal processo de manipulação realizado nos bancos de dados, pois realizando consultas é que os usuários obtêm as informações necessárias para a tomada de decisões.

Esses sistemas apresentaram uma grande evolução pois passaram a utilizar formas de interação baseadas em paradigmas visuais, tanto para apresentação do domínio de interesse quanto para construção das consultas, minimizando os conhecimentos sintáticos necessários nesta atividade.

Avaliando diversos sistemas descritos na literatura, foi possível perceber que os mecanismos implantados para melhorar a interação dos sistemas com seus usuários ainda são ineficientes para usuários que tenham pouco ou nenhum conhecimento sobre a tarefa a ser realizada ou ainda sobre o modelo semântico consultado.

A arquitetura proposta nesta tese visa construir sistemas de consultas visuais, que realizem uma cooperação com seus usuários, baseados na utilização de interfaces adaptativas, relaxamento de consultas e de paradigmas visuais. Os sistemas produzidos com esta arquitetura apresentam mecanismos de gerenciamento indireto, ou seja, realizam funções para auxiliar o usuário sem que este as tenha requisitado.

Baseado nesta arquitetura desenvolvemos um protótipo utilizando os ícones como forma de representação visual. Essa escolha foi impulsionada pelo fato dos ícones serem considerados na literatura como a forma mais eficiente de aprendizado e memorização e com isso mais indicados para facilitar a realização de consultas por usuários inexperientes.

Entretanto, somente a utilização de ícones não garante a construção de um sistema de consulta que seja eficiente para esse tipo de usuário. Outros aspectos, como a forma de operar sobre esses ícones, sua organização na interface e principalmente a

maneira de combiná-los para construção de consultas, são importantes para garantir a construção de um sistema com esta qualidade.

De acordo com a pesquisa realizada, este protótipo é o primeiro sistema de consulta que utiliza o mecanismo de interfaces inteligentes. Outros sistemas de consulta já foram desenvolvidos utilizando o relaxamento de consultas mas a adaptação de uma interface visual de consulta é uma inovação.

O uso de interfaces adaptativas, consideradas interfaces inteligentes, não vem sendo muito difundido se compararmos, por exemplo, com o número de assistentes inteligentes desenvolvidos. As razões para que isto ocorra podem ser percebidas no comentário realizado por ZWICKER e REINHARD(1990): "de modo geral, não se justifica o uso de interfaces inteligentes para sistemas de fácil utilização. Mesmo sistemas mais complexos devem ser considerados com cautela, ... Interfaces inteligentes são recursos certamente úteis, porém, para um conjunto limitado de contextos".

Avaliando este comentário, surge a pergunta: Será que os sistemas de consulta fazem parte deste contexto limitado?

A construção de consultas é uma tarefa complexa em virtude do volume de informações envolvidas na realização desta tarefa. Não só o conhecimento sobre o sistema de consulta, mas também sobre o esquema de dados da base consultada são necessários para que o usuário seja capaz de realizar suas consultas. A cada nova base de dados disponibilizada pelo sistema, o usuário necessita de novos conhecimentos. Este fato já é suficiente para validar o uso de interfaces adaptativas em sistemas de consulta, pois o processo de aquisição de novos conhecimentos pelos usuários nunca termina.

Para minimizar as desvantagens geradas com o uso de interfaces adaptativas, como a falta de previsão da interface e principalmente o sentimento de perda de controle por parte do usuário, propomos na arquitetura formas lentas de adaptação da interface, baseadas em zonas de proximidade das classes do modelo mais utilizadas pelo usuário, e por faixas de adição pré definidas.

Em nosso protótipo, as regras definidas para o sistema também consideram o tempo de compreensão, pelo usuário, das adaptações realizadas. Novas alterações só são efetuadas na interface se a última alteração dirigida pelo sistema ou pelo usuário já tiver sido realizado a n acessos anteriores, sendo n definido de acordo com outras

características do usuário, como por exemplo sua categoria, iniciante, médio ou especialista.

A principal desvantagem desta arquitetura é sua dependência em relação a um profissional que construa a base de metadados, estabelecendo as zonas de proximidade, o grau de importância dos elementos de consulta, as faixas de adição e os tópicos que irão determinar como a interface será adaptada e como o relaxamento de consulta será realizado. Se este profissional não estabelecer esses parâmetros de forma correta, o sistema pode se tornar ineficiente.

Um trabalho futuro que pode ser desenvolvido é a criação de formas automáticas para construção da base de metadados, principalmente com relação aos dados necessários à adaptação da interface e ao relaxamento da consulta. Uma possibilidade seria utilizar um agente inteligente que fosse capaz de obter essas informações, a partir de um esquema de dados.

Esta tese mostrou que é possível desenvolver sistemas de consulta utilizando a representação visual baseada em ícones e a manipulação direta e principalmente a tecnologia de interfaces adaptativas.

Para finalizar este trabalho, vamos completar as tabelas apresentadas no capítulo 2, incluindo o protótipo desenvolvido. A Tabela 6.1 apresenta as operações da linguagem de consulta implementadas pelos sistemas. A Tabela 6.2 apresenta a classificação dos sistemas de acordo com os critérios estabelecidos no capítulo 2.

Sistemas	Modelo de Dados	Operações Implementadas ou Definidas para Implementação
QBE	Relacional	relacionalmente completo.
Access	Relacional	relacionalmente completo.
Pasta'3	Relacional	seleção, projeção, junção, agregações, quantificadores.
QBD	Relacional	relacionalmente completo.
IconicBrowser	OO	seleção, projeção, junção, união, consultas aninhadas, agregações, aplicação de mensagem aos objetos.
OdeView	OO	seleção, projeção, junção.
SUPER	Relacional & OO	criação e alteração de classes e objetos; seleção, projeção, quantificador existencial.
GOODIES	OO	selecção, junção.
IRIS	ER & Relacional	relacionalmente completo.
BDE	OO	selecção, projeção.
SOPView	OO	seleção, projeção, junção, consultas aninhadas.
PESTO	OO	seleção, junção, disjunção, quantificadores.
PROTÓTIPO	OO	seleção, projeção, junção

Tabela 6.1 – Tabela de operações de consulta implementadas nos sistemas de consulta.

Sistemas	Modelo BD	Representação Visual	Compreensão do domínio	Formulação de consulta	Reuso de Consultas	Adaptação para o usuário
QBE	Relacional	Formulários	Não possuí	Casamento	não	não
Access	Relacional	Diagramas & Formulários	Não possuí	Casamento	não	não
Pasta'3	Relacional	Formulários & Diagramas	Não possuí	Casamento Subconsultas	sim	Funções de auxílio
QBD	Relacional	Diagramas	Refinamento e seleção	Navegação & Subconsultas	sim	não
IconicBrowser	OO	Ícones	Navegação	Subconsultas	não	não
OdeView	OO	Diagramas & Ícones	Navegação	Navegação	não	não
SUPER	Relacional e OO	Formulários & Diagramas	Navegação	Navegação	sim	não
GOODIES	OO	Formulários	Navegação	Navegação	não	não
IRIS	ER & Relacional	Formulários & Diagramas	Navegação	Navegação	não	não
BDE	OO	Formulários & Ícones	Não possuí	Subconsultas	sim	Funcionalidades para auxílio
SOPView	OO	Formulários, Diagramas & Ícones	Não possuí	Navegação Subconsultas	não	não
PESTO	OO	Formulários, Diagramas & Ícones	Não possuí	Navegação	sim	não
PROTÓTIPO	OO	Ícones & Diagramas	Refinamento & Navegação	Subconsultas	parcial	Adaptação da interface

Tabela 6.2 – Tabela comparativa dos sistemas de consultas visuais.

Apêndice I - Regras do Protótipo

_usuario X tem _experienciaSistema 'sim':
_usuario X tem experiencia sistema,
_usuario X efetuou _numero Z de acessos,
_numero Z menor 10,
_usuario X efetuou _numero J de consultas,
_numero J menor 10.

_usuario X tem _experienciaSistema 'sim':
_usuario X tem experiencia sistema,
_usuario X efetuou _numero Z de acessos,
_numero Z maior 10,
_usuario X efetuou _numero J de consultas,
_numero J menor 10.

_usuario X tem _experienciaSistema 'sim':
_usuario X efetuou _numero Z de acessos,
_numero Z maiorIgual 10,
_usuario X efetuou _numero J de consultas,
_numero J maiorIgual 20.

_usuario X tem _experienciaSistema 'sim':
_usuario X efetuou _numero Z de acessos,
_numero Z maiorIgual 10,
_usuario X efetuou _numero J de consultas,
_numero J maiorIgual 10,
interface do _usuario X foi melhorada _numero J vezes,
_numero J maiorIgual 3,
_usuario X aceitou _numero M de melhorias,
_numero M maiorIgual 3.

_usuario X tem _experienciaTarefa 'sim':
_usuario X nao tem experiencia tarefa,
_usuario X efetuou _numero J de consultas,
_numero J menor 15,
_usuario X realiza _numero V per cento de consultas parecidas,
_numero V menorIgual 40,
_usuario X utiliza em media _numero M de operadores,
_numero M maiorIgual 3,
_usuario X utiliza em media _numero L de classes,
_numero L maiorIgual 2.

_usuario X tem _experienciaTarefa 'sim':
_usuario X tem experiencia tarefa,
_usuario X efetuou _numero J de consultas,
_numero J maior 10,
_usuario X realiza _numero V per cento de consultas parecidas,
_numero V menorIgual 50,
_usuario X utiliza em media _numero M de operadores,
_numero M maiorIgual 3,
_usuario X utiliza em media _numero L de classes,
_numero L maiorIgual 2.

_usuario X tem _experienciaTarefa 'sim':
_usuario X efetuou _numero J de consultas,

_numero J maiorIgual 15,
_usuario X realiza _numero V per cento de consultas parecidas,
_numero V menorIgual 60,
_usuario X utiliza em media _numero M de operadores,
_numero M maiorIgual 3,
_usuario X utiliza em media _numero L de classes,
_numero L maiorIgual 2.

_usuario X e _tipo 'iniciante' :
_usuario X nao tem experiencia tarefa,
_usuario X nao tem experiencia sistema.

_usuario X e _tipo 'iniciante' :
_usuario X nao tem experiencia tarefa,
_usuario X tem experiencia sistema.

_usuario X e _tipo 'medio' :
_usuario X tem experiencia tarefa,
_usuario X nao tem experiencia sistema.
_usuario X realiza nenhum _consultas.

_usuario X e _tipo 'medio' :
_usuario X tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X realiza nenhum _consultas.

_usuario X e _tipo 'especialista' :
_usuario X tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X realiza _consultas 'complexas'.

_usuario X realiza _acessos 'frequentes' :
_usuario X tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 5,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X considera _tarefa 'importante',
_usuario X nao realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 10,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X nao considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 10,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X nao considera _tarefa 'importante',
_usuario X nao realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 20,

_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 3,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X considera _tarefa 'importante',
_usuario X nao realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 5,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X nao considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 5,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X nao considera _tarefa 'importante',
_usuario X nao realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 10,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X nao tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 1,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X nao tem experiencia tarefa,
_usuario X nao tem experiencia sistema,

_usuario X considera _tarefa 'importante',
_usuario X nao realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 2,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X nao tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X nao considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 5,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X nao tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X nao considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 5,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X nao tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 5,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X nao tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X considera _tarefa 'importante',
_usuario X nao realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 10,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X nao tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X nao considera _tarefa 'importante',
_usuario X realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 10,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X realiza _acessos 'frequentes' :
_usuario X nao tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X nao considera _tarefa 'importante',
_usuario X nao realiza _acesso 'mandatorio',
_usuario X apresenta intervalo de _numero Z dias,
_numero Z menorIgual 20,
_usuario X efetuou _numero K de acessos,
_numero K maior 10,
_usuario X efetuou _numero M de consultas,
_numero M maior 15.

_usuario X constroi _consultas 'repetitivas' :
_usuario X realiza _acessos 'frequentes',
_usuario X realiza _numero Z per cento de consultas parecidas,
_numero Z maiorIgual 60.

_usuario X constroi _consultas 'repetitivas' :
_usuario X realiza nenhum _acessos,
_usuario X realiza _numero Z per cento de consultas parecidas,
_numero Z maiorIgual 80.

_usuario X realiza _consultas 'complexas':
_usuario X realiza _acessos 'frequentes',
_usuario X tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X utiliza em media _numero Z de operadores,
_numero Z maiorIgual 5,
_usuario X utiliza em media _numero Y de classes,
_numero Y maiorIgual 5.

_usuario X realiza _consultas 'complexas':
_usuario X realiza nenhum _acessos ,
_usuario X tem experiencia tarefa,
_usuario X tem experiencia sistema,
_usuario X utiliza em media _numero Z de operadores,
_numero Z maiorIgual 4,
_usuario X utiliza em media _numero Y de classes,
_numero Y maiorIgual 5.

_usuario X realiza _consultas 'complexas':
_usuario X realiza _acessos 'frequentes',
_usuario X tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X utiliza em media _numero Z de operadores,
_numero Z maiorIgual 4,
_usuario X utiliza em media _numero Y de classes,
_numero Y maiorIgual 4.

_usuario X realiza _consultas 'complexas' :
_usuario X realiza nenhum _acessos,
_usuario X tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X utiliza em media _numero Z de operadores,
_numero Z maiorIgual 3,
_usuario X utiliza em media _numero Y de classes,
_numero Y maiorIgual 4.

_usuario X realiza _consultas 'complexas':
_usuario X realiza _acessos 'frequentes',
_usuario X nao tem experiencia tarefa,
_usuario X nao tem experiencia sistema,
_usuario X utiliza em media _numero Z de operadores,
_numero Z maiorIgual 2,
_usuario X utiliza em media _numero Y de classes,
_numero Y maiorIgual 4.

interface do _usuario X tem _area 'consultas salvas':
_usuario X realiza nenhum _acessos,
_usuario X constroi _consultas 'repetitivas'.

interface do _usuario X tem _area 'consultas salvas':

_usuario X realiza _acessos 'frequentes'.

interface do _usuario X com _classes 'mais_usados' :

_usuario X realiza _acessos 'frequentes',

_usuario X constroi _consultas 'repetitivas'.

interface do _usuario X com todas _classes:

_usuario X realiza _acessos 'frequentes',

_usuario X constroi nenhuma _consultas.

interface do _usuario X com _classes 'mais_usados' :

_usuario X realiza nenhum _acessos,

_usuario X constroi _consultas 'repetitivas'.

interface do _usuario X com _classes 'mais_importantes' :

_usuario X realiza nenhum _acessos,

_usuario X constroi nenhuma _consultas.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X constroi nenhum _consultas,

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 4,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 6,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X constroi nenhum _consultas,

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 4,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 8,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 6,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 8,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X realiza nenhum _acessos,

_usuario X constroi _consultas 'repetitivas',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 3,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 6,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X realiza nenhum _acessos,

_usuario X constroi _consultas 'repetitivas',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 3,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X realiza nenhum _acessos,

_usuario X constroi nenhum _consultas,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 6,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 6,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X realiza nenhum _acessos,

_usuario X constroi nenhum _consultas,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 6,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero W maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X efetuou _numero K de acessos,

_numero K maior 6,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 6,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X efetuou _numero K de acessos,

_numero K maior 6,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maiorIgual 4.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

_usuario X efetuou _numero K de acessos,

_numero K maior 6,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J igual 0.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 12,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 4,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X constroi nenhum _consultas,

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 8,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 4,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza nenhum _acessos,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 7,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 4,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X constroi nenhuma _consultas,

_usuario X realiza nenhum _acessos,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 10,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 4,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X efetuou _numero K de acessos,

_numero K maior 12,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 4,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 12,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X constroi nenhum _consultas,

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 8,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza nenhum _acessos,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 7,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X constroi nenhuma _consultas,

_usuario X realiza nenhum _acessos,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 10,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

_usuario X efetuou _numero K de acessos,

_numero K maior 12,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 3.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 18,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 3,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X constroi nenhum _consultas,

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 14,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 3,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza nenhum _acessos,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 13,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 3,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X constroi nenhuma _consultas,

_usuario X realiza nenhum _acessos,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 16,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 3,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X efetuou _numero K de acessos,

_numero K maior 18,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W maior 3,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 18,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X constroi nenhum _consultas,

_usuario X realiza _acessos 'frequentes',

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 14,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X constroi _consultas 'repetitivas',

_usuario X realiza nenhum _acessos,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 13,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X constroi nenhuma _consultas,

_usuario X realiza nenhum _acessos,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 16,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

_usuario X efetuou _numero K de acessos,

_numero K maior 18,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0,

interface do _usuario X adicionou classes 'mais_proximas' no _numero W de acessos anteriores,

_numero W igual 0,

o sistema realizou para o _usuario X alteracao no _numero J de acessos anteriores,

_numero J maior 2.

interface do _usuario X adicionar _classes 'faixa 1':

_usuario X e _tipo 'iniciante',

_usuario X efetuou _numero K de acessos,

_numero K maior 12,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0,

o sistema realizou para o _usuario X alteracao no _numero W de acessos anteriores,

_numero W maior 4.

interface do _usuario X adicionar _classes 'faixa 2':

_usuario X e _tipo 'iniciante',

_usuario X efetuou _numero K de acessos,

_numero K maior 12,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 6,

o sistema realizou para o _usuario X alteracao no _numero W de acessos anteriores,

_numero W maior 4.

interface do _usuario X adicionar _classes 'faixa 3':

_usuario X e _tipo 'medio',

_usuario X efetuou _numero K de acessos,

_numero K maior 18,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 10,

o sistema realizou para o _usuario X alteracao no _numero W de acessos anteriores,

_numero W maior 6.

interface do _usuario X adicionar _classes 'faixa 4':

_usuario X e _tipo 'medio',

_usuario X efetuou _numero K de acessos,

_numero K maior 24,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y igual 0.

interface do _usuario X adicionar _classes 'faixa 5':

_usuario X e _tipo 'medio',

_usuario X efetuou _numero K de acessos,

_numero K maior 24,

_usuario X realizou alteracao no _numero Y de acessos anteriores,

_numero Y maior 14,

o sistema realizou para o _usuario X alteracao no _numero W de acessos anteriores,

_numero W maior 6.

interface do _usuario X retirar _classes 'mais_proximas' :

_usuario X e _tipo 'iniciante',

interface do _usuario X adicionou classes 'mais_proximas' no _numero Y de acessos anteriores,

_numero Y maiorIgual 5,

_usuario X nao utilizou classes 'mais_proximas'.

interface do _usuario X retirar _classes 'mais_proximas' :

_usuario X e _tipo 'medio',

interface do _usuario X adicionou classes 'mais_proximas' no _numero Y de acessos anteriores,

_numero Y maiorIgual 6,

_usuario X nao utilizou classes 'mais_proximas'.

interface do _usuario X retirar _classes 'mais_proximas' :

_usuario X e _tipo 'especialista',

interface do _usuario X adicionou classes 'mais_proximas' no _numero Y de acessos anteriores,

_numero Y maiorIgual 7,

_usuario X nao utilizou classes 'mais_proximas'.

interface do _usuario X retirar _classes 'faixa 1' :

_usuario X e _tipo 'iniciante',

interface do _usuario X adicionou classes faixa 1,

_usuario X nao utilizou classes faixa 1,

interface do _usuario X adicionou classes da faixa no _numero Y de acessos anteriores,

_numero Y maiorIgual 3.

interface do _usuario X retirar _classes 'faixa 2' :

_usuario X e _tipo 'iniciante',

interface do _usuario X adicionou classes faixa 2,

_usuario X nao utilizou classes faixa 2,

interface do _usuario X adicionou classes da faixa no _numero Y de acessos anteriores,

_numero Y maiorIgual 3.

interface do _usuario X retirar _classes 'faixa 3' :

_usuario X e _tipo 'medio',

interface do _usuario X adicionou classes faixa 3,

_usuario X nao utilizou classes faixa 3,

interface do _usuario X adicionou classes da faixa no _numero Y de acessos anteriores,

_numero Y maiorIgual 3.

interface do _usuario X retirar _classes 'faixa 4' :

_usuario X e _tipo 'medio',

interface do _usuario X adicionou classes faixa 4,

_usuario X nao utilizou classes faixa 4,

interface do _usuario X adicionou classes da faixa no _numero Y de acessos anteriores,

_numero Y maiorIgual 3.

interface do _usuario X retirar _classes 'faixa 5' :

_usuario X e _tipo 'medio',

interface do _usuario X adicionou classes faixa 5,

_usuario X nao utilizou classes faixa 5,

interface do _usuario X adicionou classes da faixa no _numero Y de acessos anteriores,

_numero Y maiorIgual 3.

interface do _usuario X com _atributos 'mais_usados' :

_usuario X realiza _acessos 'frequentes',

_usuario X constroi _consultas 'repetitivas'.

interface do _usuario X com todos _atributos :

_usuario X realiza _acessos 'frequentes',

_usuario X constroi nenhuma _consultas.

interface do _usuario X com _atributos 'mais_usados' :

_usuario X realiza nenhum _acessos,

_usuario X constroi _consultas 'repetitivas'.

interface do _usuario X com _atributos 'mais_importantes' :

_usuario X realiza nenhum _acessos,

_usuario X constroi nenhuma _consultas.

interface do _usuario X com _relacionamentos 'mais_usados' :

_usuario X realiza _acessos 'frequentes',

_usuario X constroi _consultas 'repetitivas'.

interface do _usuario X com todos _relacionamentos :

_usuario X realiza _acessos 'frequentes',

_usuario X constroi nenhuma _consultas.

interface do _usuario X com _relacionamentos 'mais_usados' :

_usuario X realiza nenhum _acessos,

_usuario X constroi _consultas 'repetitivas'.

interface do _usuario X com _relacionamentos 'mais_importantes' :

_usuario X realiza nenhum _acessos,

_usuario X constroi nenhuma _consultas.

interface do _usuario X com _operadores 'mais_usados' :

_usuario X realiza nenhum _acessos,

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z maior 60.

interface do _usuario X com _operadores 'mais_importantes' :

_usuario X realiza nenhum _acessos,

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z menor 40.

interface do _usuario X com _operadores 'mais_usados' :

_usuario X realiza _acessos 'frequentes',

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z maior 60.

interface do _usuario X com _operadores 'mais_importantes' :

_usuario X realiza _acessos 'frequentes',

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z menor 40.

interface do _usuario X com _operadores 'mais_usados' :

_usuario X realiza nenhum _acessos,

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z maior 60.

interface do _usuario X com _operadores 'mais_importantes' :

_usuario X realiza nenhum _acessos,

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z menor 40.

interface do _usuario X com _operadores 'mais_usados' :

_usuario X realiza _acessos 'frequentes',

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z maior 70.

interface do _usuario X com _operadores 'mais_importantes' :

_usuario X realiza _acessos 'frequentes',

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z menor 50.

interface do _usuario X com _operadores 'mais_usados' :

_usuario X realiza nenhum _acessos,

_usuario X realiza nenhuma _consultas,

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z maior 80.

interface do _usuario X com _operadores 'mais_importantes' :

_usuario X realiza nenhum _acessos,

_usuario X realiza nenhum _consultas,

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z menor 80.

interface do _usuario X com _operadores 'mais_importantes' :

_usuario X realiza _acessos 'frequentes',

_usuario X realiza nenhum _consultas,

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z menor 70.

interface do _usuario X com _operadores 'mais_usados' :

_usuario X realiza _acessos 'frequentes',

_usuario X realiza nenhuma _consultas,

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z maior 70.

interface do _usuario X com todos _operadores :

_usuario X realiza _acessos 'frequentes',

_usuario X realiza _consultas 'complexas',

_usuario X realiza _numero Z per cento de consultas parecidas,

_numero Z menor 50.

interface do _usuario X com _operadores 'mais_importantes' :

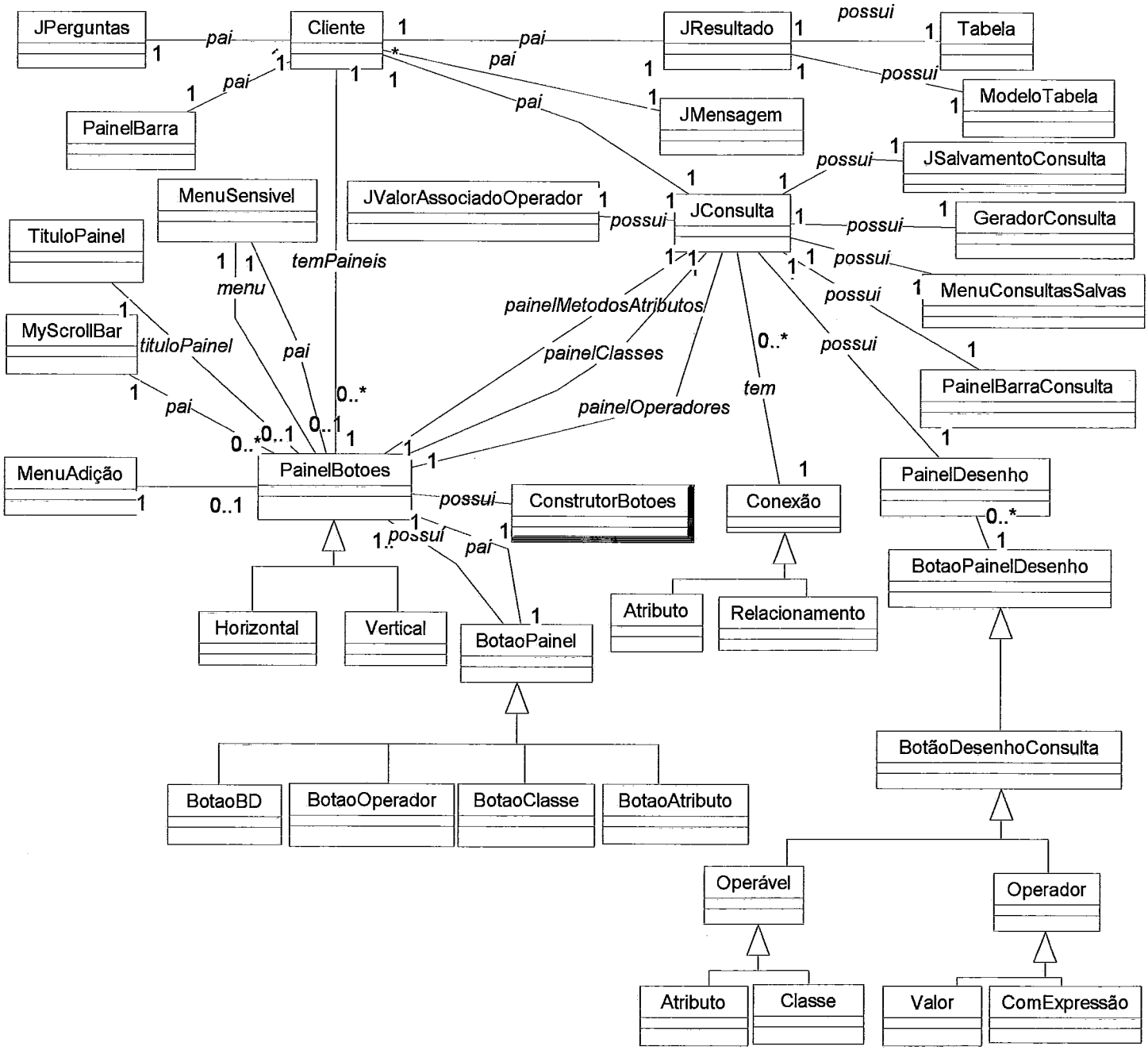
_usuario X realiza _acessos 'frequentes',

_usuario X realiza nenhum _consultas,

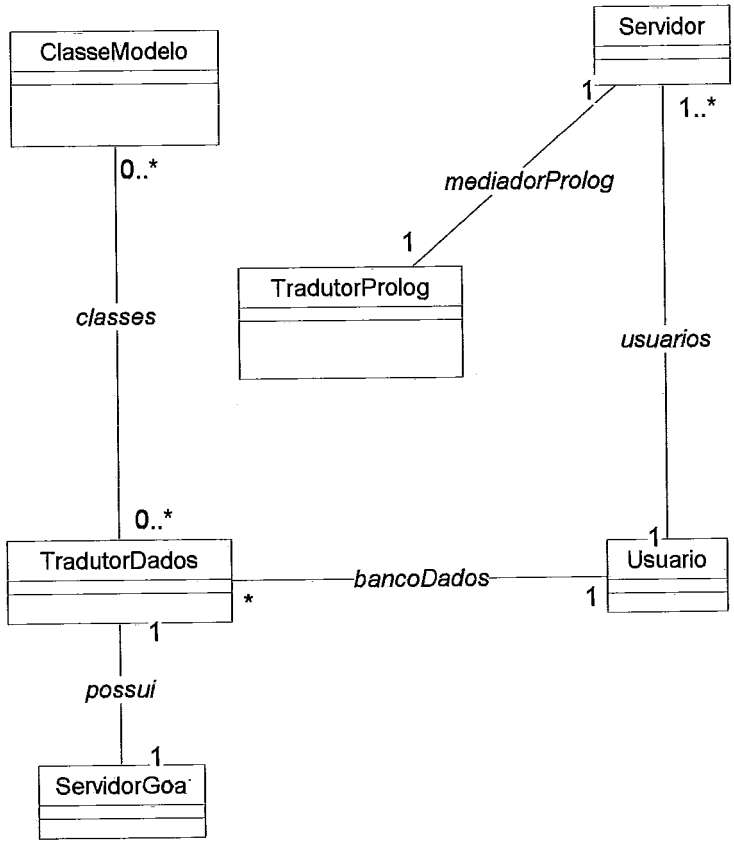
_usuario X e _tipo 'iniciante'.

Apêndice II – Modelos de Projeto

II.1 Modelo do Cliente



II.2 Modelo do Servidor



Bibliografia

A-KAUTZ, H., SELMAN, B. e COEN, M., "Bottom-Up Design of Software Agents", *Communications of the ACM*, Vol. 37, No. 7, pp. 143-147, Jul, 1994.

ABITEBOUL, S., HULL, R., VIANU, V., "Notes on Practical Languages"., *Foundations of Databases,* ed. 1, cap. 7, Addison-Wesley Publishing Company, 1995.

AGRAWAL, R., GEHANI, N. H., SRINIVASAN, J., "OdeView: The Graphical Interface to Ode". In: *Proceedings of the ACM SIGMOD*, pp. 34-43, Atlantic City, 1990.

AUDDINO, A , AMIEL, E., DENNEBOUY, Y., DUPONT, Y., FONTANA, E., SPACCAPIETRA, S., TARI, Z, "Database Visual Environments based on advanced data models", In: *Advanced Visual Interfaces*, World Scientific, pp. 156-170, 1992.

BAECKER, R., SMALL, I., MANDER, R., "Bringing Icons to Life", In: *Procceding of ACM Conference Human Computer Interface*, pp.1-6, New Orleans, Abr., 1991.

BARR, A., FEIGENBAUM, E., A., "Introduction", In: Barr, A., Feigenbaum, E., A, The Handbook of Artificial Intelligence, v. 1, ed. 1, cap. 1, California, HeurisTech Press & Willian Kaufmann Inc, 1981.

BEVILAQUA, J. S., *"Banco de Dados Estatístico Orientado a Objetos"*, Tese de M.Sc., COPPE / UFRJ, Rio de Janeiro, RJ, Brasil, 1994.

BIRD, S. D., "Toward a taxonomy of multi-agent systems", International Journal Man-Machine Studies, Vol. 39, No. 4, pp. 689-705, 1993.

BROWNE, D, TOTTERDELL, P., NORMAN, M., *Adaptative User Interface*, ed. 1, Academic Press, California, USA, 1990.

BURNS, L. M., MALHOTA, A , SOCKUT, G., WHANG, K., Y. , "AERIAL: ad-hoc entity relationship investigation and learning". *International Journal Man-Machine Studies*, v. 38, pp. 607-623, 1993.

CAREY, M., HAAS, L., MAGANTY, V., WILLIAMS, J., "PESTO: An Integrated Query/Browser for Object Databases", In: *Proceedings of the 22nd VLDB Conference*, pp. 203-214, Bombay, India, 1996.

CAROLIS, B. D., ROSIS, F. D., ERRORRE, S., "A user-adapted iconic language for the medical domain", *International Journal Human-Computer Studies*, Vol. 43, No. 4, pp. 561-578, Out, 1995.

CATARCI, T., COSTABILE, M. F., LEVIALDI, S., BATINI, C., *Visual Query Systems for Databases: A Survey*. Relatório Técnico 04.91, Universidade de Roma, 1991.

CATTEL, R. G. G., *The Object Database Standard: ODMG-93*, ed 1, Morgan Kaufmann Publishers Inc., 1994.

CATTEL, R. G. G. & BARRY, D. K. , *The Object Database Standard: ODMG-2.0*, 1 ed, Morgan Kaufmann Publishers Inc., 1997.

CHANG, S., "Principles of Visual Languages", In: Shu, N. C., *Visual Programming*, Ed. 1, Cap. 1, Los Angeles, Van Nostranol Reinhold Company, 1988.

CHANG, S. W., LEE, S. H., KIM, H. J., "SOPView: A Visual Query and Object Browsing Environment for SOP OODBMS", In:

Proceedings of th 20th. IEEE Annual International Computer Software and Applications Conference, Seol, 1996.

CHU, W. W., MERZBACHER, M. A. e BERKOVICH, L., "The Design and Implementation of CoBase", In: *Proceedings of ACM SIGMOD*, Washigton, USA, 1993.

DATE, C. J., *Introdução a Sistemas de Banco de Dados,* 4 ed., Editora Campus, 1991.

DATTOLO, A., LOIA, V., "Active distributed framework for adaptative hypermedia", *Int. Journal Human-Computer Studies*, Vol, 46, No. 5, pp. 605-627, Maio,1997.

DOWNTOM, A., *Engineering the Human-Computer Interface*, McGraw-Hill Book Company, 1992.

EBERTS, R. E., BITTIANDA, K. P., "Preferred mental models for direct manipulation and command-based interfaces", *International Journal Man-machine Studies,* Vol. 38, pp. 769-785, Jun, 1993.

EDMONDS, E. A., CANDY, L., JONES, R., SOIFI, B., "Support for Collaborative Design: Agents and Emergence", In: *Communications of the ACM*, Vol 37, No. 7, pp. 41-53, Jul, 1994.

ELMASRI, R., NAVATHE, S. B., *Fundamentals of Database Systems*, 2 ed. , The Benjamin/Cummings Publishing Company, Inc, 1994.

ESPINOZA, F. e HÖÖK, K., "A WWW Interface to na Adaptative Hypermedia System", *PAAM'96*, London, Abr, 1996.

ETZIONI, O., WELD, D. S., "Intelligent Agents on the Internet: Fact, Fiction, and Forecast, In: IEEE Expert Intelligent Systems & their Applications, Vol 10, No. 4, Ago, 1995.

FAMILANT, M. E., DETWEILER, M. C., "Iconic reference: evolving perspectives and an organizing framework", *International Journal Man-Machine Studies*, Vol. 39, pp. 705-728, Mai, 1993.

FINK, J., KOBSA, A., NILL, A., "User-Oriented Adaptivity and Adaptability in the AVANTI Project", In: *Proceedings of the Conference: Designing for the Web: Empirical Studies*, 1996.

(url: http://zeus.gmd.de/hci/projects/avant

i/publications/ms96.html).

FORSLUND, G., "Toward Cooperative Advice-Giving Systems: A Case Study in Knowledge-Based Decision Support", In: IEEE Expert Intelligent Systems & their Applications, Vol 10, No. 4, Ago, 1995.

GENTNER, D., NIELSON, J., "The Anti-Mac Interface", *Communications of the ACM*, Vol. 39, No. 8, pp. 70-82, Ago, 1996.

HÖÖK, K., "Steps to take before IUI becomes real", *Workshop "The reality of intelligent interface technology"*, Edinburgh, Mar, 1997.

HUANG, K., "Visual Interface Design Systems", In: Shu, N. C., Visual Programming, Ed. 1, Cap. 2, Los Angeles, Van Nostranol Reinhold Company, 1988.

KOBSA, A., WAHLSTER, W., *User Models in Dialog Systems,* ed. 1, USA, Springer-Verlag, 1989.

KUNTZ, M., MELCHERT, R., "Pasta-3's Graphical Query Language: Direct Manipulation, Cooperative Queries, Full Expressive power". *In: Proceedings of the Fifteenth International Conference on Very Large Data Bases*,pp.97-105, Amsterdam, 1989.

LAUREL, B.,"Interface Agents: Metaphors with Character", In: The Art of Human-Computer Interface Design, Vol. 1, Addison-Wesley Publishing Company, pp. 355-365, 1990.

LE, V. T., *Techniques of Prolog programming with implementation of logical negation and quantified goals.* 1 ed. Canberra, John Wiley & Sons Inc, 1993.

LE, V. T., *ESSLN An Expert System Shell With Logical Negation*, Relatório Técnico 90/3, Universidade de Camberra, 1990.

LUCENA, C. J. P., GUARANYS, P. Y., "New Approach to User Modelling Based on Double Stereotypes", *Journal of the Brazilian Computer Society*, Vol. 1, No. 3, Abril, 1995.

MAES, P., "Agents that Reduce Work and Information Overload", In: Communications of the ACM, Vol 37, No. 7, pp. 31-40, Jul, 1994.

MACIEJEWSKI, A. A., KANG, Y., "A Student Model of Katakama Reading Proficiency for a Japanese Language Intelligent Tutoring System", In: *IEEE Transactions on Systems. Man, and Cybernetics*, Vol. 24, No. 9, pp. 1347-1357, Set, 1994.

MARTINS, F. M., "A Software Environment for the Semi-Automatic Design of Adaptative User Interface", 2nd *ERCIM Workshop on "User Interface for All"*, Praga, Nov, 1996.

MATHE, N. e CHEN, J., "A User-Centered Approach to Adaptive Hypertext based on an Information Rwelevance Model",*Proceedings of th 4th International Conference on User Modeling(UM'94),* pp. 107-114, Hyannis, MA, Ago, 1994.

MAURO, R. C, *Aspectos de Gerência de Objetos Persistentes: A Implementação do GOA++*, Tese de MSc, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Dez, 1998.

MAYHEW, D. J., *Principles and Guidelines in Software User Interface Design*, Ed. 1, New Jersey, Prentice Hall, 1992.

MIAH, T., KARAGEORGOU, M., KNOTT, R. P., "Adaptative Toolbars: An Architectural Overview", 3$^{rd.}$ *ERCIM Workshop on "User Interfaces for All"*, França, Nov, 1997.

MITCHEL, T., CARUANA, R., FREITAG, D., MCDERMOTT J., Zabowski, D., "Experience with a Learning Personal Assistant", In: Communications of the ACM, Vol 37, No. 7, pp. 81-105, Jul, 1994.

NETTO, J. F. M., *Um tutor inteligente para o ensino de xadrez,* Tese de M.Sc. , COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Mar, 1995.

NIELSEN, J., "Noncommand user interfaces", *Communications of the ACM*, Vol. 36, No. 4, pp. 83-99, 1993.

NORMAN, D. A., "How Might People Interact with Agents", In: Communications of the ACM, Vol 37, No. 7, Jul, 1994.

OHLSSON, S., "Constraint-Based Student Modelling", *Journal of Artificial Intelligence in Education*, Vol. 3, No. 4, 1993.

OLIVEIRA, J. L. , *Browsing and Querying in Object-Oriented Databases*. Reletório Técnico DCC-12/92, UNICAMP, 1992.

PARKER, H., ROAST, C., Siddiqi, J., "Towards a Framework for Investigating Temporal Properties in Interaction", *SIGCHI Bulletin*, Vol. 29, No. 1, pp. 56-59, Jan, 1997.

RAMOS, C., Braga, J. L. e LAENDER, A. H. F., "Uma Interface Cooperativa para Consultas a Banco de Dados Relacionais", In: *Simpósio Brasileiro de banco de Dados*, pp. 113-126, São Carlos, SP, out 1996.

RIECKEN, D., "M: An Architecture of Integrated Agents", In: *Communications of the ACM*, Vol 37, No. 7, pp. 107-116, Jul, 1994.

SANTUCCI, G., SOTTILE, P. A., *Query by Diagram: A Visual Environment for Querying Databases*. Relatório Técnico 15.90, Universidade de Roma , "LA SAPIENZA", 1990.

SELKER, T., "Coach: A Teaching Agent that Learns", In: *Communications of the ACM*, Vol 37, No. 7, Jul, 1994.

SLEEMAN, D., "UMFE: A User Modelling Front-End subsystem"; International Journal Man-Machine Studies, Vol. 23, No. 1, 1985.

SOCKUT, G. H., BURNS, L. M., MALHOTRA, A., WHANG, K. , "GRAQULA: A graphical query language for entity-relationship or relational databases". *In Data and Knowledge Engineering*, Vol. 11, pp. 171-202, North-Holland, 1993.

STEPHANIDIS, C., AKOUMIANAKIS, D., SAVIDIS, A., "Design representations and development support for user interface adaptation", 1st *ERCIM Workshop on "Towards User Interface for All: Current effords and future trends"*, Grécia, Out, 1993.

SUTCLIFFE, A. G., MCDERMOTT, M., "Integrating methods of human-computer interface design with structured systems development", *International Journal of Man-Machine Studies,* Vol. 34, No. 5, pp. 631-654, Mai, 1991.

TSUDA, K., YOSHITAKA, A, HIRAKAWA, M., TANAKA, M., ICHIKAWA, T. "IconicBrowser: An Iconic Retrieval System for Object-Oriented Databases". *Journal of Visual Languages and Computing*, v. 1, pp.59-76, 1990.

VANDERDONCKT, J. M., BODART, F., "Encapsulating Knowledge For Intelligent Automatic Interaction Objects Selection", *INTERCHI*, Abr, 1993.

ZISSOS, A. Y., WITTEN, I. H., "User modelling for a computer coach: a case study", *International Journal Man-Machine Studies*, Vol. 23, No. 6, 1985.

ZLOOF, M., transparências do mini-curso: "The Evolution of Visual Tools for DBMS and Application Programming", *Simpósio Brasileiro de Banco de Dados*, São Carlos, Out, 1996.

ZWICKER, R., REINHARD, N., "Interfaces inteligentes: perspectivas para novas formas de aprendizado e uso de sistemas", Revista Brasileira de Computação,Vol. 5, No. 3, pp. 17-25, Mar, 1990.

WARN, A., *What's a Intelligent Interface ?*, notas de introdução de um seminário de Annika Warn, http://www.sics.se/~annika/ii_links.html, Mar, 1997.

WATERWORTH, J. A., CHIGNELL, M. H., ZHAI, S. M., "From icons to interface models: designing hypermedia from bottom up",

International Journal Man-Machine Studies, Vol. 39, pp. 453-472, Mar, 1993.

WIELEMAKER J., Manual de Referência do SWI-Prolog 2.9, Universidade de Amsterdam, Dept. of Social Science Informativcs(SWI), 1997.

WOOLDRIDGE, M., JENNINGS, N. R., "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review, V. 10, No. 2, pp. 115-152, Jan., 1995