

# Um Estudo sobre Ferramentas de Apoio para o Projeto de Sistemas de Banco de Dados

Nara Martini Bigolin

José Mauro Volkmer de Castilho

Instituto de Informática - UFRGS

Caixa Postal - 15064, Cep 91501-970, Porto Alegre, RS

Email: (bigolin,castilho)@inf.ufrgs.br

## Resumo

O uso de ferramentas e técnicas de Inteligência Artificial para apoiar o projeto de Sistemas de Bancos de Dados é uma área de pesquisas cujos produtos tem grande potencial comercial. Apesar disso, há poucos pesquisadores trabalhando ativamente na área. Este trabalho apresenta parte de um estudo sobre sistemas baseados em conhecimento desenvolvidos como ferramentas para apoiar e assistir projetistas de Bancos de Dados em seu trabalho. Uma coleção significativa de sistemas foi escolhida, e cada sistema está descrito aqui de forma resumida. O estudo se conclui com uma análise e uma comparação entre os sistemas do conjunto.

## Abstract

The use of tools and techniques of Artificial Intelligence to support Database Systems design activities is a research area of Computer Science whose products may have considerable commercial value. Despite this, research activity in the area is not so intense as one would expect. This work presents part of a study on knowledge based systems developed as tools to support Database design and assist DB designers in performing their task. A significant collection of systems was chosen, and each system is briefly described. The study is concluded with an analysis and comparison of the systems in the collection.

**Palavras-Chave:** projeto de banco de dados, sistemas baseados em conhecimento, processamento de linguagem natural.

# 1. INTRODUÇÃO

A Inteligência Artificial (IA) é alvo de crescente interesse interdisciplinar e já tem muita importância em várias áreas de conhecimento. Na área de Sistemas de Banco de Dados (SBD), há várias aplicações de IA, como interfaces inteligentes de banco de dados, banco de dados dedutivos, ferramentas para projeto de SBD's, etc.

A IA pode, ainda, ser empregada para gerar automática ou semi-automática, um banco de dados. Vários sistemas tem sido desenvolvidos usando tecnologia de sistemas especialistas e processamento de linguagem natural para auxiliar o desenvolvimento de projeto de banco de dados.

A tecnologia mais usada nas ferramentas desenvolvidas para auxílio ao projeto de banco de dados são Sistemas Baseados em Conhecimento, os quais são organizados de tal maneira que os fatos do domínio de conhecimento, sobre o qual o sistema trabalha, e o conhecimento sobre projeto de BD's, são separados do conhecimento geral, e o raciocínio é separado dos fatos.

Sistemas baseados em conhecimento consistem de quatro componentes:

1. Uma base de conhecimento, que contém conhecimento geral sobre o domínio e específico sobre o problema tratado.
2. Uma memória de trabalho, usada para armazenar fatos e resultados intermediários sobre o problema específico a ser resolvido.
3. Um mecanismo de inferência, que raciocina sobre as tarefas. O mecanismo contém um interpretador que decide como aplicar o conhecimento geral da base de conhecimento para um caso específico.
4. Interface de usuário, que é usada para interagir com o usuário quando informações adicionais são solicitadas ou quando resultados devem ser comunicados.

O Processamento da Linguagem Natural, uma área bastante ativa de Inteligência Artificial, também tem sido usado em pesquisas e estudos ligados ao uso e desenvolvimento de SBD. Entre estes estudos destacam-se os que propõem o uso de interfaces de consulta em linguagem natural (LN) [SAB 91] [TOM 87] [WIN 84], e os que propõem o tratamento formal de requisitos expressos em LN obtidos durante as primeiras fases do projeto do banco de dados [ARI 92] [LOH 91] [MET 93].

O ciclo de desenvolvimento de um projeto de banco de dados geralmente é dividido em quatro partes: Coleta e Análise de Requisitos do Sistema; Projeto Conceitual (formalização dos requisitos coletados na primeira fase); Projeto Lógico (determina a forma de implementação); Projeto Físico (implementação propriamente dita).

Como parte da investigação sobre a aplicação de técnicas de IA ao projeto de SBD's, estudou-se vários trabalhos na área, os quais usam metodologias e tecnologias diversas para tratar as fases do

desenvolvimento de projeto de banco de dados. Entre as metodologias, destaca-se a DATAID [CER 83] [BAT 92], para a qual foram desenvolvidas técnicas e ferramentas de apoio para todas as etapas de desenvolvimento do projeto de banco de dados.

Na próxima seção serão descritos oito sistemas de IA desenvolvidos para apoiar o desenvolvimento do projeto de sistemas de banco de dados, para apresentar um quadro atual do conjunto das ferramentas deste tipo. Na seção 3 é feita uma discussão sobre tais sistemas e na última seção são apresentadas considerações finais e pesquisas futuras.

## 2. SISTEMAS PESQUISADOS

Os sistemas descritos neste item são divididos em três grupos conforme as fases do projeto de banco de dados: No primeiro grupo o está Intelligent Interview System e o View Creation System que auxiliam no desenvolvimento da fase de Análise de Requisitos; No segundo grupo se encontra duas ferramentas (o Assistente Especialista e o Assistente Projetista) desenvolvidas em Universidades no Brasil, além do Modeller e do AVIS que foram desenvolvidos em universidades do exterior, todas elas auxiliam na segunda fase do projeto de banco de dados (Projeto Conceitual). No último grupo está o SECSI e o GESDD que auxiliam da fase de Projeto Lógico.

### 2.1 Intelligent Interview System (IIS) [KAW 86]

Foi desenvolvido para extrair informações (requisitos) através de entrevista com o usuário de um banco de dados. O sistema pode aprender sobre diferentes aplicações e sobre o processo de entrevistas, bem como entrevistar diferentes usuários. O IIS entrevista usuários e constrói um modelo do domínio da aplicação do usuário chamado Estrutura de Plano. A Estrutura de Plano é semelhante ao modelo E-R e consiste em um conjunto de planos, onde cada plano representa uma atividade e corresponde a um relacionamento. Cada plano é obtido dos requisitos do usuário e é representado através de *Frames*.

Esta ferramenta apresenta os seguintes componentes:

**Parser:** O parser transforma os requisitos do usuário em linguagem natural (língua inglesa) para um plano temporário.

**Gerador:** Transforma o plano temporário em sentenças para fazer a verificação. Após, o IIS cria frames com operadores, os quais correspondem aos verbos da entrada fornecida pelo usuário, e os coloca em sua base de conhecimento, preenchendo cada *slots* dos *frames* com informações apropriadas, fornecidas pelo usuário. O *Frame* representado neste plano é armazenado numa memória de pequenos termos. A seguir será descrita a base de conhecimento e as informações armazenadas nesta memória.

**Base de Conhecimento:** A base de conhecimento do IIS é dividida em 4 categorias:

1. Há um dicionário para cada domínio de aplicação do sistema.

2. A estratégia de perguntas serve para extrair informações fornecidas pelo usuário.

3. Conhecimento do planejamento é usado para determinar se o sistema tem informações que podem ser aplicadas no plano.

4. Conhecimento do projeto de esquema conceitual é usado para transformar a estrutura de plano em um esquema conceitual do projeto de banco de dados. Assim, a estrutura de plano é decomposta em operadores, onde cada operador é transformado numa relação.

**Memória:** O sistema tem uma memória de pequenos termos, uma memória de trabalho e uma lista de aplicação. A memória de pequenos termos é usada para guardar a estrutura de plano do problema. A memória de trabalho armazena regras de produção que são usadas em: Estratégia de perguntas; Conhecimento do Planejamento e Conhecimento do projeto do esquema conceitual.

**Interface do Usuário:** é baseado em uma entrada textual onde contém uma estratégia de perguntas para obter informações do usuário.

A entrada do IIS é um conjunto de requisitos em linguagem natural (língua inglesa) provido pelo usuário com a estrutura "Eu quero saber...", e a saída é um projeto de banco de dados relacional parcial. O sistema foi implementado em Data General MV/8000 II e escrito em C-Prolog.

## 2.2 View Creation System [STO 86]

É um sistema especialista que foi desenvolvido para deduzir requisitos obtidos do usuário para um banco de dados. O sistema comunica-se com o usuário de um banco de dados através de um diálogo projetado para obter entidades, relacionamentos e atributos das informações fornecidas pelo usuário. O usuário não precisa conhecer técnicas ou terminologia do projeto de banco de dados. Durante a sessão, o sistema apresenta os conceitos de entidades, relacionamentos e atributos num breve tutorial. O usuário é conduzido a descrever sua aplicação usando a estrutura do modelo E-R, enquanto o sistema tenta detectar e resolver inconsistências, ambigüidades e redundâncias das informações do usuário.

O View Creation System alerta ao usuário para entidades e relacionamentos que aparecem implícitos, mas que não foram especificados pelo usuário. O sistema também é capaz de detectar quando o usuário usa estruturas erradas, por exemplo, representa alguma entidade quando ela pode ser atributo de outra entidade.

Este sistema possui os seguintes componentes:

**Base de Conhecimento:** Consiste de aproximadamente 500 regras. Estas regras são obtidas inicialmente dos conceitos de projeto de banco de dados. A base de conhecimento é então expandida para incorporar perícia (conhecimento específico), a qual é obtida pela inclusão da experiência do

projetista de banco de dados através de uma consulta, na qual ele fornece informações para uma aplicação hipotética. O sistema foi testado formalmente em aproximadamente 8 aplicações de banco de dados.

**Mecanismo de Inferência:** O conhecimento do sistema é representado por um conjunto de regras procedurais e de produção. Regras procedurais controlam a ordem em que as atividades do projeto são carregadas. Regras de produção foram obtidas através do conhecimento do especialista.

**Interface do Usuário:** É baseada em uma entrada textual que permite ao usuário pedir uma exposição do conjunto corrente de requisitos.

A entrada é um diálogo entre o usuário e o sistema, no qual o usuário fornece os requisitos do banco de dados para o sistema e a saída é um modelo de dados relacional representando os requisitos do usuário. O modelo E-R para estes requisitos é uma saída intermediária. O sistema é escrito em Arity Prolog e roda em um PC.

### 2.3 Um Assistente Especialista Para Aquisição de Requisitos [ARI 92]

É um assistente especialista para auxiliar na fase de especificação, usando para apoiar a modelagem E-R. A função do assistente é auxiliar o projetista durante a fase de especificação, através de um sistema de perguntas e respostas.

O assistente é composto de uma interface em linguagem natural, uma base de dados, uma base de conhecimento e uma máquina de inferência.

**Interface em linguagem natural:** é um ambiente, no qual o usuário fornece os conceitos informais da aplicação, através de um diálogo em um sub-conjunto da Linguagem Natural, guiado pelo assistente.

**Base de Conhecimento:** onde é armazenado o conhecimento sobre a metodologia de especificação (neste caso, Modelo E-R).

**Base de Dados:** onde é armazenado o conhecimento, obtido através do diálogo com o usuário e, de onde se obterá, posteriormente, o modelo textual.

**Máquina de Inferência:** é a parte do assistente, que consulta a base de conhecimento, para validar a especificação, segundo o modelo E-R.

A interface é responsável pela interação entre o usuário e o assistente, a qual é feita através de um diálogo guiado pelo assistente em um sub-conjunto de linguagem natural. As respostas são transformadas com a ajuda de uma gramática DCG (Definite Clause Grammar) em primitivas do Modelo E-R e armazenadas numa base de dados.

O assistente, usando a máquina de inferência, analisa estas primitivas e com um conhecimento específico armazenado dentro da base de conhecimento, verifica possíveis erros e incomplezas. Após, gera para o conteúdo da base de dados uma especificação semi-formal baseada no modelo E-R.

A entrada do Assistente Especialista são textos representando diálogos em linguagem natural (língua portuguesa), fornecidos pelo usuário e a saída é um modelo (E-R) textual correspondente a especificação semi-formal. O sistema foi implementado em C-Prolog.

#### 2.4 Assistente Projetista [BIG 94]

Foi desenvolvido para auxiliar o projetista na fase de Modelagem Conceitual do projeto de Banco de Dados, traduzindo a descrição de requisitos expressos em linguagem natural coletados na fase de Análise de Requisitos para uma especificação formal segundo o modelo de Entidade-Relacionamento (E-R). Tem como componentes: um Analisador Morfológico, um Dicionário Morfológico, um Gerador de Dicionário e um Analisador Sintático Ascendente.

**Analisador Morfológico:** permite analisar morfológicamente uma frase em linguagem natural, usando um dicionário morfológico e "modelos" que configuram e classificam as palavras da língua portuguesa, fornecendo como saída, uma a uma, as palavras da frase original, cada uma com suas informações morfológicas, para que o tradutor de frases possa fazer o reconhecimento das estruturas das frases em linguagem natural, e assim executar a tradução correta [BIG 91].

**Gerador de Dicionários:** gera um dicionário para o analisador sintático com as palavras contidas no texto fornecidas pelo analisador morfológico. O gerador de dicionário recebe do analisador morfológico as palavras analisadas com a respectiva classificação morfológica e as transforma em "fatos" do Prolog.

**Analisador Sintático Ascendente:** Gera a árvore sintática das frases de entrada e as mapeia para componentes de um modelo conceitual (modelo E-R), usando regras heurísticas de mapeamento.

**Interface do Usuário:** aceita uma entrada textual dos requisitos do sistema de banco de dados.

As interfaces de entrada e saída foram desenvolvidas com uma ferramenta para gerar interfaces gráficas chamada Guide [SUN 90], disponível em ambiente Unix. A interface de entrada é um editor de texto e a interface de saída é um editor gráfico.

O Analisador Morfológico e o Gerador de Dicionário foram implementados em C em ambiente UNIX. O Analisador Sintático e as regras heurísticas foi implementadas em Prolog. A entrada do Assistente Projetista é formada por requisitos expressos em linguagem natural em forma de texto, e a saída é um diagrama E-R.

#### 2.5 Modeller [TAU 89]

É um sistema especialista que produz um modelo de dados conceitual expresso em modelo E-R estendido. O usuário fornece a descrição de uma empresa (requisitos) em linguagem natural (neste caso, língua inglesa) e o Modeller transforma estas informações em um conjunto de declarações sobre entidades, relacionamentos e atributos.

O usuário pode solicitar ao sistema o modelo conceitual inteiro para fazer os devidos ajustes semânticos.

O sistema é composto das seguintes partes:

**Base de Conhecimento:** A base de conhecimento do Modeller é baseado no modelo E-R estendido e taxionomias introduzidas por pesquisadores. Modeller possui duas taxionomias principais - uma que classifica conceitos do modelo E-R estendido e outra, a taxionomia de fatos. O nível de conceitos mais alto na primeira taxionomia são as entidades, relacionamentos, propriedades, domínios e restrições. Na outra taxionomia, fatos são classificados em dois tipos: o fato-modelo é um componente do modelo conceitual sendo desenvolvido, enquanto que um fato ambiente é informação sobre o estado do processo de modelagem. Modeller distingue entre estes fatos fornecidos pelo usuário e aqueles gerados pelo sistema.

Modeller tem um conjunto de regras *default*, que permitem preencher espaços no modelo conceitual. Por exemplo, se a cardinalidade de um relacionamento não é conhecida, o sistema coloca o valor 0.

Muitos fatos de uma empresa podem ser concluídos através da análise de sua estrutura. Modeller possui um conjunto de regras causais que permitem:

1. detectar e corrigir violações de integridade;
2. conferir a consistência entre a estrutura e componentes do modelo conceitual;
3. gerar *triggers* do banco de dados.

**Interface do Usuário:** A interface do usuário aceita entrada textual, com a possibilidade de uma interface gráfica para uma versão futura.

A entrada da ferramenta consiste em textos com requisitos sobre uma empresa, que serão convertidos em declarações sobre entidades, relacionamentos e atributos, e a saída é uma representação E-R estendida do modelo conceitual da empresa. O sistema foi implementado em Arity Prolog numa estação de trabalho Compaq 386/20.

## 2.6 Automated View Integration System - AVIS [WAG 89]

AVIS objetiva a integração de informações do usuário em um esquema conceitual simples. AVIS examina duas visões ao mesmo tempo para detectar coisas em comum e resolver inconsistência. Os tipos de inconsistências que podem ocorrer são nomes, construções (isto é, entidades versos relacionamentos), contexto (isto é, entidade e atributo versos relacionamentos e atributos, dependendo do contexto) e conflitos de significado.

O sistema AVIS possui os seguintes componentes:

**Base de Conhecimento:** AVIS inclui conhecimento necessário para: a. Reconhecimento de conflitos entre as visões; b. Resoluções de conflitos e, c. Sequência da busca e procedimentos com soluções.

**Interface do Usuário:** O AVIS usa várias janelas, mas nem todas permanecem na tela simultaneamente. As janelas tem a seguinte função: Integração de visões; Consulta proposta pelo AVIS para o usuário; Resposta do usuário; Atividades pendentes; Conclusão de fatos com a tarefa de integração.

A entrada do AVIS é um conjunto de visões do usuário representado em modelos E-R. A saída é um esquema conceitual em E-R, representando a integração de todas as visões de entrada. O sistema foi escrito em Turbo Prolog e roda em PC.

## 2.7 *Système en Conception de Système d'Informations* - SECSI [BOU 85]

SECSI foi projetado como uma ferramenta inteligente para auxiliar o usuário no projeto de banco de dados. O sistema inicia com uma descrição simples de uma aplicação do usuário e então comunica-se, através de um diálogo, com o usuário para obter as informações necessárias para um projeto lógico do banco de dados que é representado por rede semântica. É composto de regras, fatos, máquina de inferência escrita em Prolog, interfaces e uma metabase para armazenar a saída. As regras apresentam a metodologia do projeto do banco de dados, enquanto que o fato descreve a aplicação do usuário.

Este sistema foi desenvolvido para interagir com dois tipos de usuário: o projetista de banco de dados e o usuário final. O usuário pode fornecer uma descrição inicial do problema a ser projetado no banco de dados, de várias formas: linguagem formal, linguagem declarativa, um sub-conjunto restrito da linguagem natural ou uma interface gráfica.

O SECSI é composto das seguintes partes:

**Base de Conhecimento:** O sistema é baseado em diferentes tipos de conhecimento como: teorias para a modelagem relacional; teorias para a modelagem semântica; algumas regras específicas e experimentais relatada pelo usuário e o conhecimento de domínio específico, como conhecimento sobre medicina, negócio bancária, etc. Regras e restrições são representadas por regras de produção, enquanto que uma rede semântica representa fatos sobre a aplicação do usuário.

**Redes Semântica:** Uma rede semântica representa um problema de banco de dados do usuário. Algumas restrições são acrescentadas aos arcos e nodos para representar a semântica da aplicação.

**Regras do Projeto:** As regras do projeto consistem na violação da consistência e regras de transformação estrutural, que funcionam sobre a rede semântica para manter a consistência do esquema conceitual e transformá-lo em um esquema relacional normalizado; no conhecimento geral, como os tipos dos nodos e dos arcos que podem aparecer na rede semântica; e na hierarquia de meta-regras que controlam a seqüência dos passos do projeto.

**Interface do Usuário:** a entrada da ferramenta possui várias formas como: entrada textual; entrada gráfica ou através de uma linguagem formal.

A entrada do SECSI são requisitos do usuário expressos de várias formas e a saída é um conjunto de relações na Quarta Forma Normal representando o projeto lógico. O sistema está escrito em Prolog e roda em Multics.

## **2.8 Generalized Expert System for Database Design - GESDD [DOG 89]**

O GESDD é atualmente composto por dois sistemas especialistas - o Expert System for Generating Methodologies (ESGM) e o Expert System for Database Design (ESDD). O ESGM é uma ferramenta para projetista do Banco de dados que serve para especificar novas metodologias de projeto (ou modificar as já existentes) para as fases de análise de requisitos, projeto conceitual e projeto lógico. ESDD é usado em conjunção com uma das metodologias para produzir um projeto de banco de dados, iniciando com a fase de análise de requisitos e produzindo um esquema lógico (modelo: hierárquico, de rede ou relacional).

### **a. Expert System for Generating Methodologies - ESGM**

No ESGM, o projetista de banco de dados pode especificar uma metodologia em termos de conjunto de conceitos, campos e regras e seus inter-relacionamentos. Os Conceitos são os objetos básicos que são divididos em cada fase do processo do projeto. Para a metodologia de análise de requisitos, os objetos do mundo real, são exemplos de conceitos. Para a metodologia do projeto conceitual baseado no modelo E-R, os conceitos relevantes são entidades, atributos e relacionamentos. Os conceitos do projeto lógico dependem da maneira como o modelo de dados lógico é usado. Conceitos relevantes para o modelo hierárquico são registros e associação pai-filho. Similarmente, relação é um conceito no modelo relacional. Um campo é uma propriedade de um conceito e tem um único nome.

### **b. Expert System for Database Design - ESDD**

O ESDD funciona da seguinte maneira:

1. Chamada do ESDD: O projetista de banco de dados entra com os requisitos em frames ou em uma área de trabalho que contém a base de conhecimento do sistema. O ESDD não fornece nenhum auxílio para análise dos requisitos do usuário.

2. Manutenção dos frames: Se um novo banco de dados é projetado, uma metodologia de projeto é selecionada. (Modificações em projetos existentes também podem ser feitas. O processo do projeto é carregado de acordo com as fases de análise de requisitos, projeto conceitual e projeto lógico, como especificado na seleção da metodologia.

3. Fase de Análise de Requisitos: A análise dos requisitos é feita em termos de conceitos que foram definidos para esta fase pelo ESDD. A lista de conceitos é mostrada em um menu. Após, o projetista seleciona uma delas e uma forma é apresentada, a qual será usada para entrar uma instância deste conceito. As instâncias podem ser acrescentadas, removidas ou modificadas e o ESDD trata a consistência e confere cada operação. Exemplificando: se um elemento é acrescentado, o sistema assegura que ele seja único.

4. Fase do Projeto Conceitual: Durante o projeto conceitual, os conceitos definidos na fase de análise de requisitos são mapeados para os conceitos do esquema conceitual, seguindo as regras de mapeamento. Então o esquema é conferido para que esteja sintaticamente correto. Aspectos semânticos de um esquema conceitual podem ser checados interativamente através da execução de uma opção testa-executa. Simultaneamente, operações de adição, deleção e modificação são efetuadas pelo ESDD para detectar definições conflitantes.

5. Fase do Projeto Lógico: A fase de projeto lógico recebe o esquema conceitual e as regras de transformação do meta-esquema para selecionar o modelo lógico. A saída é um projeto de banco de dados relacional, hierárquico ou de rede e um conjunto de transações. As transações estão na forma procedural, e incluem adição, deleção, modificação e teste de comandos.

Os ESGM é composto pela seguintes partes:

**Base de Conhecimento:** O ESGM pode ser visto como sistema especialista para construção de Sistemas de Banco de Dados. Sua base de conhecimento inclui os modelos das três fases do projeto de banco de dados e a noção de conceitos, campos e regras. As metodologias do projeto são introduzidas pelo projetista de banco de dados e salvas em uma biblioteca.

**Interface do Usuário:** O projetista é guiado por um processo de especificação de metodologias através de vários menus.

O ESDD possui as seguintes partes:

**Base de Conhecimento:** O conhecimento usado pelo ESDD consiste de uma metodologia de biblioteca mantida pelo ESGM e especificações da aplicação fornecidas pelo projetista do banco de dados.

**Interface do Usuário:** O projetista do banco de dados interage com o sistema através de vários menus.

A entrada do ESGM é o conhecimento de um especialista de uma metodologia de projeto de banco de dados e a saída é um representação armazenada desta metodologia. A entrada do ESDD é um conjunto de requisitos fornecidos pelo projetista do banco de dados e a saída é um esquema lógico do banco de dados em um dos modelos de dados conhecido. O sistema foi escrito em Prolog e implementado num Burroughs A9F system.

### 3. ANÁLISE DOS SISTEMAS

Será feita uma comparação e, após, uma discussão sobre os sistemas, os quais serão separados em 3 grupos. O primeiro grupo é formado pelos sistemas ( Intelligent Interview System e View Creation System) que tratam a fase de análise de requisitos. Os sistemas Assistente Projetista, Assistente Especialista, Modeller e AVIS estão no segundo grupo e tratam a fase de modelagem conceitual e na última fase se enquadram o SECSI e o GESDD, os quais tratam a fase de projeto lógico.

Sistemas	Fases do Proj.	Usuário	Descrição	Entrada	Saída
IIS	Análise de Requisitos Proj. Conceitual Proj. Lógico	Usuário Final	Sistemas de entrevista baseado em frames.	Requisitos do usuário expressos em Inglês.	Relacionamento representado pela 1NF.
View Creation System	Análise de Requisitos; Projeto Conceitual; Proj. Lógico		Sistema interativo para visões do usuário	Requisitos fornecido pelo usuário via diálogo.	Conjunto de Relações na 4FN.
Assistente Especialista	Projeto Conceitual	Projetista	Gera um esquema ER textual a partir de requisitos do usuário.	Requisitos do usuário.	Esquema ER textual.
Assistente Projetista	Projeto Conceitual	Projetista	Cria um esquema ER a partir de requisitos expressos em Linguagem Natural.	Requisitos fornecido através de diálogo.	Esquema ER.
Modeller	Projeto Conceitual	Projetista	Cria um Esquema ER a partir da descrição do usuário de uma aplicação.	Requisitos em forma de texto fornecido pelo usuário.	Esquema Conceitual (Esquema E-R estendido).
AVIS	Projeto Conceitual	Projetista	Automatiza a transformação de visões individuais de usuários em um esquema conceitual global.	Visões de usuário individual em forma de ER.	Esquema Conceitual em forma de ER.
SECSI	Projeto Lógico	Especialista Projetista	Ajuda o projetista na transformação da descrição de uma realidade em um projeto lógico.	Requisitos do usuário representado em diversas formas.	Rede Semântica; Relação na 4FN; Restrições.
GESDD	Projeto Conceitual; Projeto Lógico	Especialista Projetista	Ferramenta para o projetista especificar metodologia do projeto de banco de dados.	Especificação de uma metodologia de projeto de BD pelo projetista.	Biblioteca de terminologias de Projeto de Banco de Dados Projeto Lógico.

### 3.1 Intelligent Interview System X View Creation System

Os dois sistemas tratam a fase de análise de requisitos, que objetiva obter informações do usuário para um banco de dados. A dificuldade, desta fase, está na dificuldade do usuário em expressar as informações necessárias. O IIS possui um domínio restrito e sua eficiência é limitada, enquanto que View Creation System possui um domínio independente e com mais eficiência, além de permitir a um usuário responder perguntas sobre uma determinada aplicação sem entender sobre terminologias de banco de dados. Outra característica é que este sistema apresenta ao usuário um tutorial sobre os conceitos de banco de dados. Um dos objetivos futuros do IIS é criar e organizar uma memória de pequenos termos para permitir a retenção de conhecimentos obtidos de cada domínio de aplicação. Isso permitiria ao sistema aprender a partir da experiência do projetista, e conseqüentemente, aumentar sua eficiência.

Sistemas	Base de Conhecimento	Efetividade	Usuário	Trabalhos Futuros	Domínio
IIS	Teoria de BD; Conhecimento de pesquisador	Limitado	Usuário com alguns conhecimento de BD.	Desenvolver uma interface onde o sistema responda perguntas do usuário.	Restrito
View Creation System	Teoria de BD; Conhecimento de especialista	Alguns testes	Usuário com alguns conhecimento de BD.	Construir uma base de conhecimento do mundo real para que o sistema automaticamente infira requisitos do usuário.	Independente do domínio

### 3.2 Assistente Especialista X Assistente Projetista X Modeller X AVIS

O problema da fase de projeto conceitual é a escolha do modelo conceitual. O Assistente Projetista possibilita trabalhar com um grande domínio, visto que tem grandes dicionários. Outro ponto, é o uso de interfaces gráficas como entrada e saída da ferramenta.

Em contrapartida, o Assistente Especialista usa uma base de conhecimento o que permite trabalhar com mais precisão sobre o domínio em questão. Porém estes dois sistemas ainda estão em fase inicial, visto que há poucas pesquisas neste assunto aqui no Brasil. Portanto, muito pode ser desenvolvido, como construir uma interface interativa durante a modelagem dos requisitos do sistema, permitir o uso de frases mais complexas para descrever os requisitos, entre outros.

O Modeller assume que um projeto conceitual claro pode ser construído considerando todos os requisitos de uma vez. O AVIS, por outro lado, produz um projeto conceitual global pela integração individual das visões do usuário. O Modeller tem como usuário o projetista, e na base de conhecimento contém a teoria de banco de dados. Já o AVIS tem como usuário o Administrador de Dados e na base de conhecimento, integração de visão. Todos possuem o domínio independente. Como trabalhos futuros para auxiliar na fase de projeto conceitual temos: Sistemas que são assistentes

para o projetista de banco de dados, que inclui checagem de elementos individuais para garantir que eles sejam incluídos num processo de integração e checagem de representação múltipla para a mesma coisa. Ou ainda, sistemas que incorporam heurísticas que testam onde o projeto conceitual está completo e consistente.

Sistemas	Base de Conhecimento	Efetividade	Usuário	Trabalhos Futuros	Domínio
Assistente Especialista	Metodologia de especificação.	Alguns testes	Projetista	Tratar ambiguidades dos requisitos.	Restrito
Assistente Projetista	Não tem.	Alguns testes	Projetista	Criar um interface interativa com o usuário	Independente do domínio
Modeller	Teoria de BD; Taxonomias.	Raciocínio sofisticado	Projetista	Testar e expandir as taxonomias.	Independente do domínio
AVIS	Integração de visões.	Teste de pesquisador	Administrador de dados	Cria novas regras para resolver conflitos	Independente do domínio

### 3.3 SECSI X GESDD

A dificuldade do projeto lógico está em assegurar que um conjunto de regras consistentes e completas existam para efetuar a tradução do modelo conceitual para o lógico. No SECSI, a base de conhecimento contém regras e heurísticas para representar explicitamente restrições que estão explícita no modelo conceitual, mas muitas vezes são perdidos na tradução para o modelo lógico, porém pode ser incluída restrições de integridade como parte da saída da ferramenta baseado em conhecimento. Na fase de projeto lógico há muito a ser feito, como: interfaces que usam terminologias de projeto de banco de dados; sistemas que possibilitam interagir com múltiplos usuários. Técnicas para integração de visões garantem a integridade das informações e podem incluir um mecanismo para detectar inconsistências e erros entre informações obtidas de múltiplos usuários.

Sistemas	Base de Conhecimento	Efetividade	Usuário	Trabalhos Futuros	Domínio
SECSI	Teoria de BD; Redes Semânticas.	Efetivo	Especialista Projetista	Criar uma interface que use terminologia de BD.	Independente do Domínio
GESDD	Conhecimento do especialista de BD.	Nenhum teste feito.	Especialista Projetista	Mecanismo de detecção de erros na entrada de múltiplos usuários.	Independente do Domínio

#### 4. CONCLUSÕES E PESQUISAS FUTURAS

Este trabalho apresentou um levantamento de diversas ferramentas desenvolvidas para auxiliar o desenvolvimento do projeto de sistemas de banco de dados. A tecnologia utilizada para desenvolver estas ferramentas é a de Sistemas Baseados em Conhecimento. Foi apresentada uma breve descrição dos componentes e do funcionamento de cada ferramenta, bem como uma comparação entre elas, e pesquisas futuras de cada uma.

No Brasil, há pouca coisa feita nesta área, visto que foram desenvolvidas apenas duas ferramentas [ARI 92] [BIG 94], com arquitetura bastante simples. Mas acredita-se que a partir destes trabalhos, possa-se desenvolver sistemas mais robustos e futuramente comercializáveis.

Portanto, pretende-se com este trabalho apresentar uma visão do que existe na área e incentivar novas pesquisas neste assunto, pois é uma área de grande importância, uma vez que em outros países já existem ferramentas comercializáveis, como é o caso da França.

#### BIBLIOGRAFIA

[ARI 92] ARI, Cecilia I. S. Um Assistente especialista para aquisição de requisitos. Campinas: DCC-UNICAMP, 1992. 189p. Dissertação de Mestrado.

[BAT 92] BATINI, C. et al. Conceptual database design: an entity-relationship approach. Redwood City: The Benjamin/ Cummings, 1992. 470 p.

[BIG 91] BIGOLIN, Nara M. et al. Sistema Gerenciador de Dicionários. Porto Alegre: PUC-RS, 1991. 57p. Trabalho de Conclusão de Graduação.

[BIG 94] BIGOLIN, Nara M. Aplicações de técnicas de processamento de linguagem natural no apoio ao projeto de banco de dados. Porto Alegre:CPGCC-UFRGS. 117p. Dissertação de Mestrado (defesa prevista para abril/94).

[BOU85] BOUZEGHOUB, M. et al. Database design tools: an expert system approach. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES, 8., may 1985, Los Altos, 32p.

[CER 83] CERI, S. Methodology and tools for database design. Amsterdam: North-Holland, 1983. x+255p.

[CHO85] CHOUBINCH, J. Form driven conceptual data modelling. Arizona: Dept. Management Information Systems, University of Arizona. 1985, 385p. Tese de Doutorado.

[DOG 89] DOGAC, B. et al. A Generalized expert system for database design. IEEE Transactions Software Engineering. v.15, n.4, p.479-491, Apr. 1989.

[KAW 86] KAWAGUCHI N. et al. An intelligent interview system for conceptual desing of database. ECAI. v.2, n.3, 1986

[LOH 91] LOH, S. Uma Linguagem comum entre usuários e analistas para definição de requisitos de sistemas de informação. Porto Alegre: CPGCC UFRGS, 1991. 198p. Dissertação de Mestrado.

[MET 93] MÉTAIS, E. et al. Application of natural language techniques to database design. In: EAST-WEST AI CONFERENCE: FROM THEORY TO PRACTICE. 7-9 Set. 1993, Moscou. 7p.

[SAB 91] SABBAGH, S. An Application of entity-relationship models to a natural language user interface: SESAME. In: INTERNATIONAL CONFERENCE DE ENTITY RELATIONSHIP APPROACH THE CORE OF CONCEPTUAL MODELLING, 9., 8-10 Out. 1991, Lausane. p.319-31.

[SCH 78] SCHNEIDER, H. et al. Report of the codasyl data description language committee. Information System. new York, v.3, n.4, p.247-320, jan. 1978.

[STO 86] STOREY, V. C. View Creation: an expert system for database design. Vancouver: Faculty of Commerce and Business - Universidade of British Columbia, 1986. 345p. Tese de Doutorado.

[SUN 90] SUN MICROSYSTEMS. Sun user's Guide: Doing More. Mountain View: Sun Microsystems. Manuais, 1990. 84p.

[TAU 89] TAUZOVICH. An Expert System for Conceptual Data Modelling. In: INTERNATIONAL CONFERENCE ON ENTITY-RELATIONSHIP APPROACH, 8., 18-20 oct. 1989, Toronto, CA.

[TOM 87] TOMA, T.; SAFTOIN, E. Natural language query for databases. Romania: [s.n], 1987. p.237-45.

[WIN 84] WINSTON, P. H; PRENDERGAST, K. A. The IA business: The Commercial uses of artificial intelligence. New York: MIT, 1984.