

UNIVERSIDADE FEDERAL DA PARAIBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

DESCONT-THM: UM SISTEMA PARA O DESENVOLVIMENTO DE
ESQUEMAS CONCEITUAIS DE DADOS EM THM (TEMPORAL
HIERARCHIC DATA MODEL) BASEADO EM FRAMES.

OSÉ LAURINDO CAMPOS DOS SANTOS

CAMPINA GRANDE
SETEMBRO - 1988

JOSÉ LAURINDO CAMPOS DOS SANTOS

DESCONT-THM: UM SISTEMA PARA O DESENVOLVIMENTO DE
ESQUEMAS CONCEITUAIS DE DADOS EM THM (TEMPORAL
HIERARCHIC DATA MODEL) BASEADO EM FRAMES.

Dissertação apresentada ao Curso de
MESTRADO EM SISTEMAS E COMPUTAÇÃO da
Universidade Federal da Paraíba, em
cumprimento às exigências para
obtenção do Grau de Mestre.

AREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO.

RAIMUNDO HAROLDO DO CARMO CATUNDA

Orientador

JOSÉ HAMURABI N. MEDEIROS

Co-Orientador

CAMPINA GRANDE

SETEMBRO - 1988

DIS
00141 23/08/88
12372



DESCONT-THM : UM SISTEMA PARA O DESENVOLVIMENTO DE
ESQUEMAS CONCEITUAIS DE DADOS EM THM (TEMPORAL
HIERARCHIC DATA MODEL) BASEADO EM FRAMES.

JOSE LAURINDO CAMPOS DOS SANTOS

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DE PÓS-GRADUAÇÃO
EM SISTEMAS E COMPUTAÇÃO DA UNIVERSIDADE FEDERAL DA PARAÍBA COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
MESTRE EM CIÊNCIAS (MSc).

Aprovada por:

RAIMUNDO HAROLDO DO CARMO CATUNDA

Presidente

JOSE HAMURABI N. MEDEIROS

Co-Orientador

ORION DE OLIVEIRA SILVA

Examinador

MARTA ELEONORA T. P. CHIANCA

Examinadora

CAMPINA GRANDE

SETEMBRO - 1988



S237d Santos, Jose Laurindo Campos dos
Descont-thm : um sistema para o desenvolvimento de esquemas conceituais de dados em THM (temporal hierarchic data model) baseado em frames / Jose Laurindo Campos dos Santos. - Campina Grande, 1988.
152 f. : il.

Dissertacao (Mestrado em Sistemas e Computacao) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Sistemas de Informacoes Geograficas 2. Dissertacao I. Catunda, Raimundo Haroldo do carmo, Prof. II. Medeiros, Jose Hamurabi N., Prof. III. Universidade Federal da Paraiba - Campina Grande (PB) IV. Título

CDU 004.4'27(043)

DESCONT-THM : UM SISTEMA PARA O DESENVOLVIMENTO DE
ESQUEMAS CONCEITUAIS DE DADOS EM THM (TEMPORAL
HIERARCHIC DATA MODEL) BASEADO EM FRAMES.

JOSE LAURINDO CAMPOS DOS SANTOS

Dissertação Aprovada em 05/09/88

R. H. do C. Catunda
RAIMUNDO HAROLDO DO CARMO CATUNDA

Orientador

Jose Hamurabi N. Medeiros
JOSE HAMURABI N. MEDEIROS

Co-Orientador

Orion de Oliveira Silva
ORION DE OLIVEIRA SILVA

Componente da Banca

Marta E. T. P. Chianca
MARTA ELEONORA T. P. CHIANCA

Componente da Banca

CAMPINA GRANDE

SETEMBRO - 1988

A Waneide Campos dos Santos.

In Memoriam.

A G R A D E C I M E N T O S

Agradeço aos meus orientadores, professores Raimundo Haroldo do Carmo Catunda e José Hamurabi N. Medeiros por suas contribuições ao meu desenvolvimento acadêmico, pela seleção dos itens e pontos mais importantes que foram desenvolvidos neste trabalho, pelas críticas construtivas e pelo paciente apoio dispensado durante todo o tempo, dando a este trabalho significado especial.

A Sidinéia Amadio, que não poupou esforços e muito incentivou, participou e sobre tudo, me fez acreditar na realização deste trabalho.

A Salomé Navarro pelas discussões, estímulo e colaboração substancial nos ajustes e sugestões realizadas durante as várias fases deste trabalho.

Ao Instituto Nacional de Pesquisas da Amazônia (INPA), pelas condições que permitiram a realização desta dissertação de mestrado.

Aos professores Marcus C. Sampaio e Ulrich Schiel, pela amizade, incentivo e idéias que se transformaram em soluções de alguns problemas surgidos.

Aos amigos Marcelo e Marta Chianca, pelo inestimável apoio e franquia de suas instalações e laboratório.

Aos amigos, pelas palavras de incentivo recebidas nas diversas fases do trabalho, tornando esse empreendimento bem mais agradável.

A todos os que integram as secretarias de Pós-Graduação e do Departamento de Sistemas e Computação UFPB/COPIN/DSC, pela fineza no tratamento de nossas necessidades acadêmicas durante o curso.

S u m á r i o

1.	INTRODUÇÃO.....	1
2.	PARTICULARIDADES DE UM PROJETO DE BANCO DE DADOS (BD).....	6
2.1	Objetivos de um projeto de BD.....	6
2.2	Fases de um projeto de BD.....	7
2.3	Benefícios resultantes de um projeto ideal.....	11
3.	MODELO DE DADOS: UMA TAXONOMIA.....	13
3.1	Modelos de dados primitivos.....	13
3.2	Modelos de dados clássicos.....	13
3.3	Modelos de dados semânticos.....	15
3.3.1	Extensões diretas dos modelos clássicos.....	15
3.3.2	Modelos matemáticos.....	16
3.3.3	Modelos indecomponíveis.....	17
3.3.4	Modelos semânticos hierárquico-estáticos.....	18
3.3.5	Modelo semântico hierárquico-dinâmicos.....	19
3.4	Modelo de dados semânticos de propósitos especiais.....	19
4.	O MODELO SEMÂNTICO DE DADOS TEMPORAL HIERÁRQUICO.....	21
4.1	Introdução.....	21
4.2	Taxonomia do modelo temporal hierárquico (THM).....	21
4.3	Conceitos básicos.....	23
4.4	Conceitos dos relacionamentos hierárquicos entre classes.....	25
4.5	Modelagem do tempo.....	26

4.6	Aspectos dinâmicos.....	29
5.	O SISTEMA DESCONT-THM.....	32
5.1	Arquitetura do DESCONT-THM.....	33
5.2	Interface amigável.....	34
5.3	Representação do conhecimento.....	35
5.3.1	Redes semânticas e frames : uma abordagem conceitual.....	37
5.3.1.1	Redes semânticas.....	37
5.3.1.2	Dedução nas redes semânticas.....	38
5.3.1.3	Frames.....	42
5.3.2	A RC do modelo semântico de dados THM baseado em frames.....	48
5.3.3	Descrição semântica do EDFRAME : o editor de frames do BC.....	51
5.4	Módulo de explanação.....	56
5.5	Módulo de inferência.....	57
5.5.1	O método depth-first informado.....	58
5.5.2	Encadeamento forward e backward.....	60
6.	DESCRIÇÃO DA ESTRUTURA DE DADOS E EXEMPLO DE UMA MODELAGEM COM VISÕES DAS ESTRUTURAS INTERNAS DAS TABELAS.....	62
6.1	A estrutura de dados do DESCONT-THM.....	62
6.1.1	Tabela de endereçamento de tarefas (Tabrefas).....	63
6.1.2	Tabela de senha de usuários (Tabsenha).....	64
6.1.3	Tabela de formato (Tabform).....	64
6.1.4	Tabela de conteúdo (Tabcon).....	65
6.1.5	Tabela de acesso aos frames (Tabaccess).....	66
6.1.6	Tabela de descrição (Tabdescr).....	67

6.1.7	Tabela para dados modelados (Tabdat).....	68
6.1.8	Tabela de frames (Tabfrm).....	69
6.1.9	Tabela de explanação (Tabhlp).....	69
6.1.10	Tabela conjunto de futuras soluções (Tabfs).....	70
6.1.11	Tabela soluções eleitas (Tabmeta).....	71
6.2	Exemplo de uma modelagem e sua representação interna.....	71
6.2.1	Modelando classes.....	79
6.2.2	Modelando relacionamentos.....	85
6.2.2.1	Estabelecendo generalizações.....	86
6.2.2.2	Estabelecendo agregações.....	88
6.2.2.3	Estabelecendo agrupamentos.....	90
6.2.2.4	Estabelecendo relacionamentos comuns.....	92
7.	CONCLUSÕES E FUTUROS TRABALHOS.....	101
	REFERÊNCIAS BIBLIOGRÁFICAS.....	103
APÊNDICE A:	Sintaxe da linguagem de definição de dados LDD/THM.....	107
APÊNDICE B:	Sintaxe da linguagem de manipulação de dados LMD/THM.....	109
APÊNDICE C:	Formalismo gráfico do modelo THM.....	113
APÊNDICE D:	Descrição das tabelas de esquemas conceituais de dados.....	115
APÊNDICE E:	Tutorial do sistema DESCNT-THM.....	123
APÊNDICE F:	Mensagens exibidas pelo sistema DESCNT-THM.....	147

Lista de Figuras

1.1 : Fases de um projeto de BD em THM.....	4
2.1 : Arquitetura de um sistema de BD.....	9
2.2 : Tipos de usuários de um BD.....	11
4.1 : Taxonomia do modelo THM.....	22
4.2 : Representação gráfica do esquema conceitual de dados de um sistema de controle de veículo em um departamento de trânsito.....	28
5.1 : Arquitetura do DESCONT-THM.....	35
5.2 : Arcos especiais para conexão de nós.....	38
5.3 : Herança de propriedade.....	39
5.4 : Organização hierárquica permitindo distribuição de propriedades.....	40
5.5 : Dedução por unificação.....	41
5.6 : Fragmento a ser unificado com o BC.....	41
5.7 : Instanciamento como resultado da inferência submetida ao BC.....	42
5.8 : Um sistema de frames.....	43
5.9 : Inferência natural.....	45
5.10: Detalhes associados aos frames e slots.....	47
5.11: Integração de regras à técnica de frames.....	48
5.12: Exemplo de distinção entre slots.....	50
5.13: Exemplo de integridade semântica do BC.....	51
5.14: Arquitetura do EDFRAME.....	53
5.15: Taxonomia de acesso aos frames.....	54
5.16: Teclas de comandos do EDFRAME.....	55

6.1	: Estrutura do registro de endereçamento de tarefas.....	63
6.2	: Estrutura do registro da tabela de senha.....	64
6.3	: Estrutura dos elementos da tabela de formato.....	64
6.4	: Estrutura dos registros da tabela de conteúdo.....	66
6.5	: Estrutura da tabela de acesso aos frames.....	66
6.6	: Estrutura da tabela de descrição.....	67
6.7	: Estrutura de uma tabela para dados modelados.....	68
6.8	: Estrutura de uma tabela de frame.....	69
6.9	: Estrutura de uma tabela de explanação.....	70
6.10	: Estrutura da tabela conjunto de soluções.....	70
6.11	: Estrutura do registro da tabela de soluções eleitas.....	71
6.12	: Esquema conceitual de dados para um sistema telefônico simples.....	73
6.13	: Organização dos dados na TABSENHA.....	73
6.14	: Organização dos dados na TABCON.....	74
6.15	: Organização dos dados na TABFORM.....	74
6.16	: Organização dos dados na TABACCESS.....	75
6.17	: Organização dos dados na TABDESCR.....	75
6.18	: Organização dos dados na TABFS.....	75
6.19	: Organização dos dados na TABMETA.....	76
6.20	: Dados armazenados na tabela CLASSE.DAT.....	80
6.21	: Dados armazenados na tabela TP_DOMI.DAT.....	81
6.22	: Dados armazenados na tabela COMPOSTO.DAT.....	82
6.23	: Dados armazenados na tabela ESTÁTICO.DAT.....	82
6.24	: Dados armazenados na tabela DINÂMICO.DAT.....	83
6.25	: Dados armazenados na tabela GENER.DAT.....	86
6.26	: Dados armazenados na tabela NOMPAPEL.DAT.....	87

6.27:	Dados armazenados na tabela	L_SUBCLA.DAT.....	87
6.28:	Dados armazenados na tabela	TP_PAPEL.DAT.....	87
6.29:	Dados armazenados na tabela	PARAM.DAT.....	88
6.30:	Dados armazenados na tabela	AGREG.DAT.....	89
6.31:	Dados armazenados na tabela	LCLACOMP.DAT.....	89
6.32:	Dados armazenados na tabela	AGRE_EXE.DAT.....	90
6.33:	Dados armazenados na tabela	AGRUP.DAT.....	91
6.34:	Dados armazenados na tabela	CLA_ELEM.DAT.....	91
6.35:	Dados armazenados na tabela	AGRU_EXP.DAT.....	91
6.36:	Dados armazenados na tabela	RELAC.DAT.....	93
6.37:	Dados armazenados na tabela	CARDMIN.DAT.....	94
6.38:	Dados armazenados na tabela	CARDMAX.DAT.....	94
6.39:	Dados armazenados na tabela	PRE_REL.DAT.....	95
6.40:	Dados armazenados na tabela	TIP_CM.DAT.....	95
6.41:	Dados armazenados na tabela	RCLACLA.DAT.....	96
6.42:	Dados armazenados na tabela	CHAVE.DAT.....	96
6.43:	Estrutura de dados que representam as construções arquiteturais do exemplo apresentado.....		100
A1	: Notação da sintaxe da LDD/THM.....		108
E1	: Menu principal do DESCNT-THM.....		125
E2	: Funções de manutenção do DESCNT-THM.....		126
E3	: Mapa geral dos arquivos para geração.....		127
E4	: Continuação do mapa dos arquivos.....		127
E5	: Gerando senhas para o BC-THM.....		128
E6	: Início do processo navegacional para acessar frames.....		129
E7	: Tela de acesso ao frame.....		130
E8	: Edição de um frame.....		130

E9	: Exemplo de subníveis não terminais do frame classe.....	131
E10	: Grau de conhecimento do usuário.....	131
E11	: Funções para modelagem do EC-THM.....	132
E12	: Modelando uma classe.....	133
E13	: Identificando os tipos de uma classe.....	133
E14	: Uma explanação solicitada.....	134
E15	: Grau de satisfação para politica de nível.....	134
E16	: Funções da modelagem de relacionamentos.....	135
E17	: Modelando generalizações.....	135
E18	: Identificando um papel.....	136
E19	: Modelando agregações.....	137
E20	: Identificando classes elementos.....	138
E21	: Modelando relacionamento entre classes.....	139
E22	: Obtendo cardinalidade.....	139
E23	: Preservando valores anteriores.....	140
E24	: Modelando relacionamentos temporais.....	141
E25	: Associando tempo às classes.....	141
E26	: Especificando tempo localizado ou quantidade de tempo.....	142
E27	: Alteração no parâmetro de tempo.....	142
E28	: Relatório da tabela de classe.....	143
E29	: Relatório da tabela de agrupamentos.....	144
E30	: Relatório da tabela de agregações.....	144
E31	: Relatório da tabela de relacionamentos temporais pré-pós....	144
E32	: Relatório da tabela de generalização-papéis.....	144
E33	: Relatório da tabela de relacionamentos.....	145
E34	: Relatório de diagnóstico das classes modeladas.....	146

DESCONT-THM : UM SISTEMA PARA O DESENVOLVIMENTO DE
ESQUEMAS CONCEITUAIS DE DADOS EM THM (TEMPORAL
HIERARCHIC DATA MODEL) BASEADO EM FRAMES.

R E S U M O

Neste trabalho é apresentado o projeto e a implementação do SISTEMA DESCONT-THM, ferramenta escrita em PASCAL e orientada ao desenvolvimento dos aspectos estáticos da modelagem de esquemas conceituais de dados para o sistema THM (Temporal Hierarchic Data Model).

Essa ferramenta permite a obtenção automática do esquema conceitual de dados de uma aplicação (universo de discurso) para posterior mapeamento ao esquema de dados relacional, dando continuidade ao desenvolvimento das fases de implementação do sistema THM.

O sistema mantém interação amigável com o usuário, proporcionando captação semântica fiel a padronização do modelo THM. Para isso utiliza o modelo de frames (quadros) para representar o conhecimento sobre o modelo THM, necessário ao processo de inferência que é realizado pelo sistema.

DESCONT-THM : A SYSTEM TO DEVELOPMENT THE CONCEPTUAL
DATA SCHEME IN THM (TEMPORAL HIERARCHIC DATA
MODEL) BASED ON FRAMES.

A B S T R A C T

This work presents the design and implementation of the DESCONT-THM SYSTEM, a tool written in Pascal and oriented towards the development of static aspects of the modeling of conceptual data schemes for the systems THM (Temporal Hierarchic Data Model) systems.

This tool automatically gives the conceptual data scheme of an application (universe of discourse) for a post-mapping to the relational data scheme, giving continuity to the development of the implementation phases of the THM system.

The system maintains a friendly interaction with the user, with a semantic capture accurate to the THM model standart. For this it uses a frame model to represent the knowledge about the THM model, necessary to the inference processing made by the system.

DESCONT-THM : UM SISTEMA PARA O DESENVOLVIMENTO DE
ESQUEMAS CONCEITUAIS DE DADOS EM THM (TEMPORAL
HIERARCHIC DATA MODEL) BASEADO EM FRAMES.

1. INTRODUÇÃO.

Em 1984 foi concluída no Instituto de Informática da Universidade de Stuttgart, Alemanha Ocidental, uma tese de doutoramento [SCHI84] intitulada: " Um Modelo Semântico de Dados para Esquemas Conceituais e sua Transformação para Esquemas Internos Relacionais " de autoria de Ulrich Schiel, cujo modelo semântico formalizado para o mapeamento foi o temporal hierárquico, também desenvolvido pelo autor.

As características deste modelo proporcionaram a idealização de um projeto de Banco de Dados (BD) em THM (proposta de transformação do modelo THM para o modelo relacional) onde cada módulo da configuração geral do sistema THM comporta técnicas específicas, a cada etapa do projeto.

Na figura 1.1 será mostrada a abrangência do trabalho ora descrito como componente de um projeto do esquema conceitual, onde são indicadas as fases de um projeto de BD baseado no modelo semântico de dados THM.

A configuração geral encontra-se subdividida nas seguintes etapas:

- (a) projeto do esquema conceitual;
- (b) verificação dos efeitos colaterais;
- (c) sistema de eventos e "triggers";
- (d) mapeamento do esquema conceitual;
- (e) interface PASCAL/SQL.

O projeto do esquema conceitual consiste na abordagem formal de um sistema de aplicação, composto de um sub-sistema responsável pela modelagem dos aspectos estáticos, denominado de DESCONT-THM descrito na presente dissertação, e um outro sub-sistema, cuja responsabilidade é a de modelagem dos aspectos dinâmicos de uma aplicação, intitulado SISDECO-THM, em fase de desenvolvimento.

A definição do esquema conceitual será realizada de forma amigável (interativa), através desses dois sub-sistemas que substituem, com recursos satisfatórios, a metodologia descrita em [MS85], e como produto final do trabalho aqui descrito, será gerado o esquema conceitual de dados.

A verificação dos efeitos colaterais [FERR87] é realizada, utilizando-se o esquema de operações existente. A verificação consiste em saber se as operações mantêm a semântica estrutural, que foi definida anteriormente no esquema conceitual de dados. Caso uma operação apresente erro semântico, o sistema de efeitos colaterais incluirá declarações a esta operação e, com isto, estará garantindo a integridade dos dados tornando-o consistente com um esquema.

As operações e transações em um BD são chamadas de eventos. Um evento é composto por um predicado que especifica em qual condição o evento torna-se verdadeiro. Neste caso, uma ou mais ações são disparadas por "triggers" que executam as transformações no sistema, correspondentes ao evento.

A automatização dos eventos previstos proporciona recurso adicional, podendo com isso gerar soluções e fornecer subsídios para formação de novos eventos, evitando que procedimentos não sejam efetuados. Em [TURA87] é descrito o sistema de eventos e "triggers".

O mapeamento do esquema conceitual é dividido em duas etapas: o mapeamento do esquema conceitual de dados para um esquema de dados relacional equivalente, e o mapeamento do esquema conceitual de operações para as operações THM executáveis.

Um arquivo será gerado contendo a codificação de comandos SQL, cujo objetivo é o de criar relações que foram geradas pelo mapeamento do esquema conceitual de dados. Um sistema de gerenciamento de banco de dados relacional acessará este arquivo e criará o esquema relacional.

Todas as operações THM executáveis serão transformadas em procedimentos PASCAL. Esta fase do projeto é detalhada em [CUNH88].

A interface PASCAL/SQL permitirá a implantação do BD com o esquema de dados relacional; e a manipulação do BD é feita através da ativação das operações THM executáveis (representação do esquema conceitual de operações).

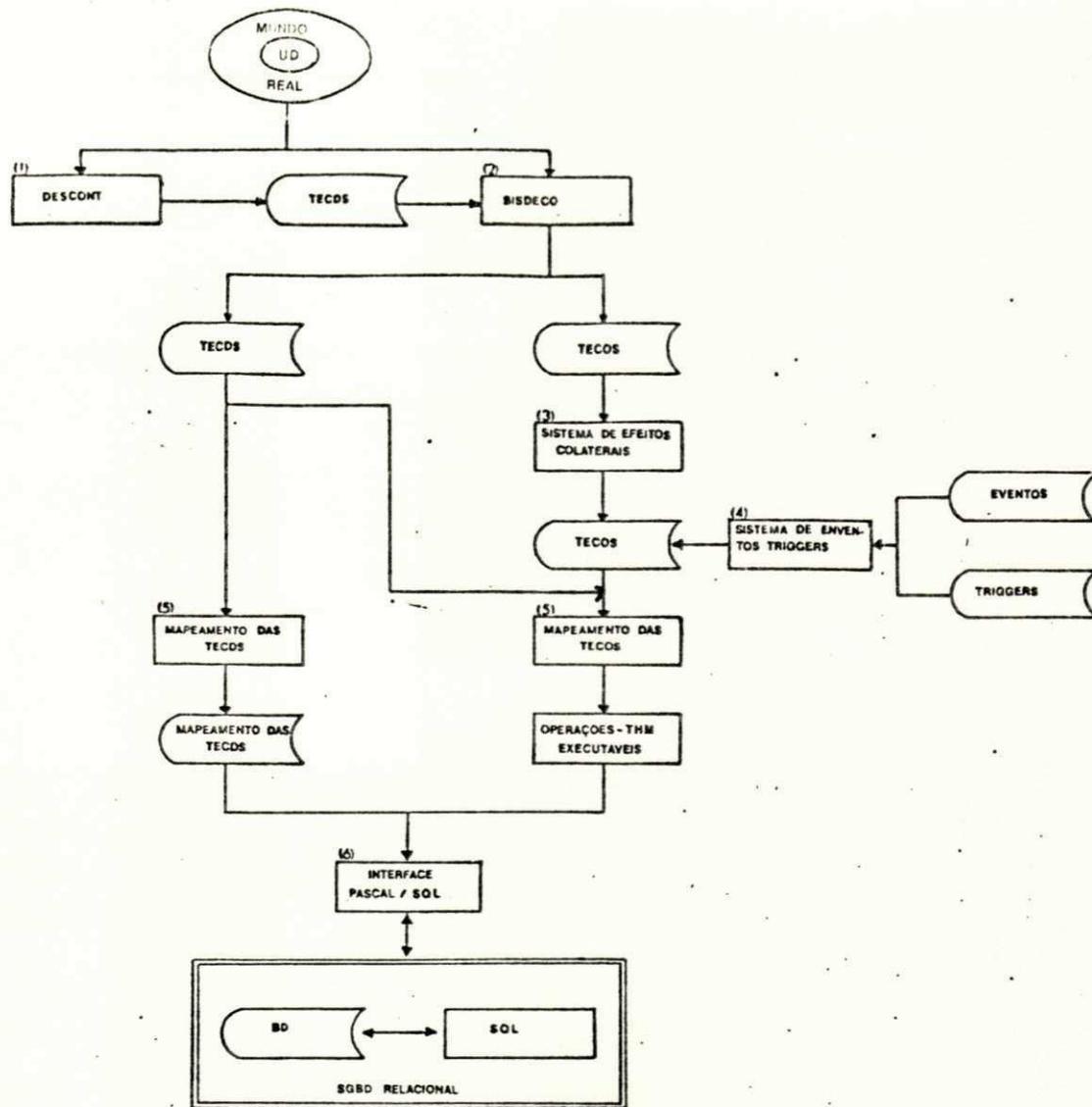


Figura 1.1 : Fases de um projeto de BD em THM.

Este trabalho tem duas finalidades principais: a primeira, permitir aos usuários do modelo THM utilizar as facilidades atribuídas ao sistema DESCONT-THM, através de interface amigável a qual, por ser apoiada em técnicas de Inteligência Artificial (IA), oferece opção diferente da linguagem de definição de dados LDD/THM ou de metodologias tradicionais, [HSY85,MS85] cuja finalidade é a mesma (obter o esquema

conceitual de dados); a segunda, auxiliar no aprendizado do modelo THM, através dos níveis de abordagem na explanação dos conceitos do THM.

O CAPÍTULO 2 apresenta particularidades e objetivos das fases de um projeto de BD convencional, classificando os usuários de BD's e identificando os benefícios resultantes de um projeto ideal.

O CAPÍTULO 3 apresenta panorama sumário das várias gerações de modelos de dados; são classificados e os aspectos que distinguem os modelos afins são ressaltados.

O CAPÍTULO 4 refere-se á apresentação do modelo semântico de dados THM com nível de detalhamento suficiente á compreensão do restante do trabalho.

O CAPÍTULO 5 contém descrição e arquitetura do DESCONT-THM, com ênfase na representação do conhecimento do modelo THM, o tipo de inferência utilizada durante os procedimentos da modelagem e suas características básicas.

O CAPÍTULO 6 apresenta a estrutura interna adotada para o sistema e um exemplo do armazenamento dos valores instanciados na estrutura interna, incluindo a descrição da gerência dessas estruturas.

O CAPÍTULO 7 resume as principais conclusões obtidas no desenvolvimento deste trabalho, além de propor algumas sugestões em termos de continuidade do mesmo.

2. PARTICULARIDADES DE UM PROJETO DE BANCO DE DADOS (BD).

2.1 Objetivos de um projeto de BD.

Os avanços tecnológicos na área de informática ocorrem de forma rápida, proporcionando como resultado imediato, as sofisticações nos processos automatizados de uma organização [FS86].

A análise dos objetivos considera os seguintes aspectos:

- (1) Propriedades das fases do desenvolvimento de um projeto de BD [HUBB81], a saber:
 - (a) elevar a qualidade dos projetos de BD;
 - (b) diminuir o tempo de desenvolvimento nos projetos de BD.

- (2) Objetivos a serem alcançados com o desenvolvimento de um projeto de BD, os quais segundo [DATE81] devem ser:
 - (a) redução de redundâncias;
 - (b) diminuição da inconsistência;
 - (c) compartilhamento dos dados;
 - (d) possibilidade de padronização;
 - (e) manutenção da integridade;
 - (f) balanceamento das necessidades conflitantes;
 - (g) independência de dados.

2.2 Fases de um projeto de BD.

As fases de um projeto de BD, apresentada na figura 2.1 [DATE81] referem-se à organização das informações, usando a abordagem da " Standards Planning and Requirements Committee" do "American National Standards Committee on Computer and Information Processing " [ANSI/X3/SPARC], que sugere considerar os três esquemas seguintes:

- (a) **esquema externo** - corresponde à representação, do ponto de vista de cada tipo de usuário, sobre o universo de discurso das informações. Tais informações só podem ser manipuladas através de operações determinadas e inerentes a cada esquema externo.

Um esquema externo não deve conflitar com os esquemas conceituais ou com qualquer outro esquema externo modelado, devendo haver total compatibilidade entre as estruturas lógicas e os tipos de SGBD's existentes [FS86, ATRE80].

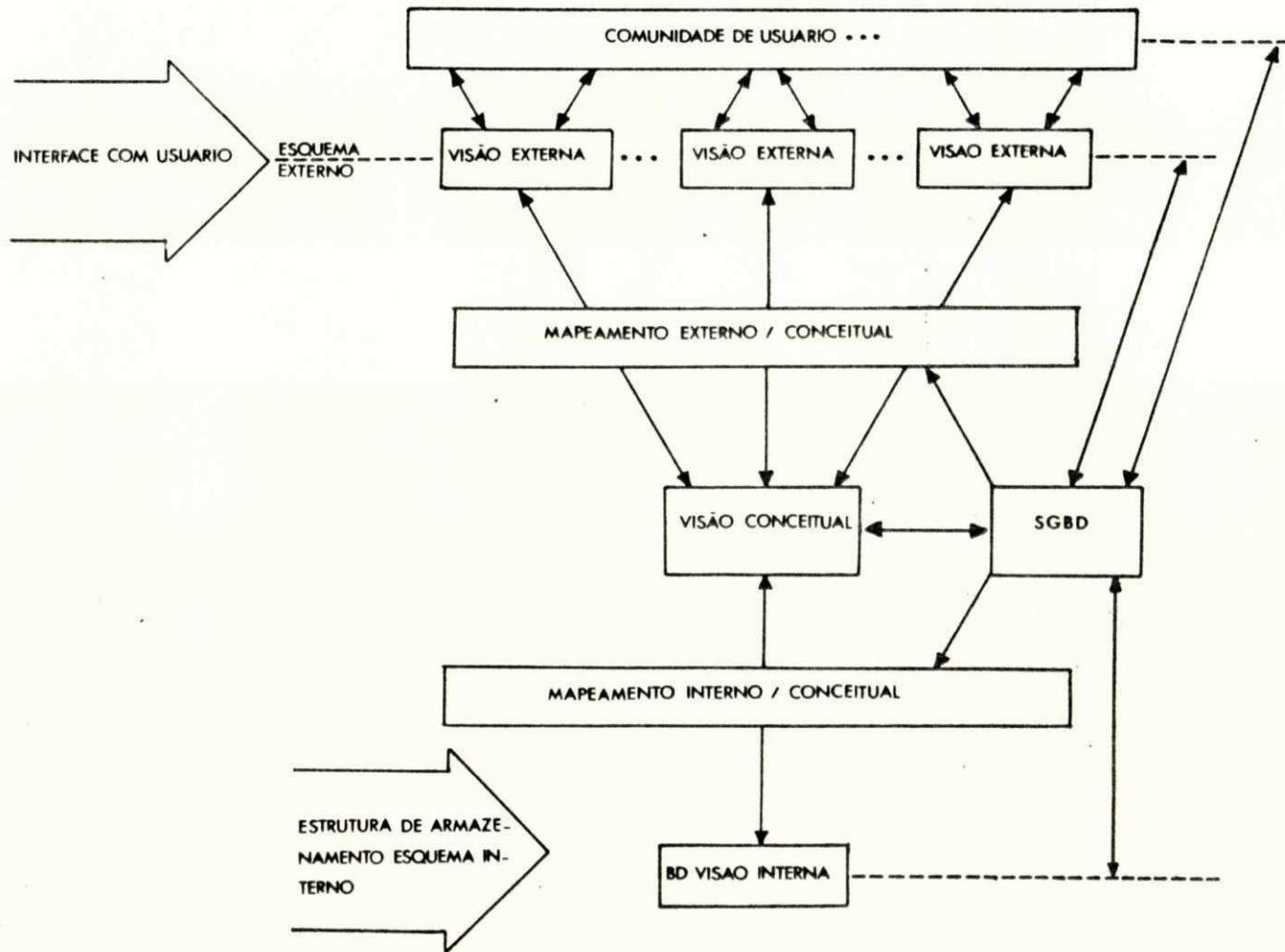
- (b) **esquema conceitual** - corresponde à representação geral de uma parte do mundo real (universo de discurso), visão esta que influenciará diretamente no universo de abrangência do BD.

No esquema conceitual, a modelagem não deve ser conjugada com as características de software e de

hardware, como também, não deve tratar de aspectos relacionados com a eficiência, memória etc. A preocupação máxima é tanto para evidenciar a prioridade existente na conceituação de entidades, atributos e relacionamentos, quanto na dissociação entre o esquema conceitual e os sistemas de gerenciamento de BD (SGBD) [FS86].

- (c) **esquema interno** - corresponde à representação da configuração do armazenamento e da manipulação de informações. O esquema interno depende da escolha tanto de um modelo de dados quanto de um SGBD. Tais escolhas são necessárias porque uma organização terá que decidir entre usar um dos sistemas disponíveis ou desenvolver um próprio que atenda às características específicas diferenciadas de um sistema de propósito geral [FS86,HUBB81].

Figura 2.1 : Arquitetura de um sistema de BD.



Para o bom desempenho de todas as estruturas que são projetadas ao longo das fases de um projeto de BD, é essencial possibilitar o uso de um número considerável de linguagens, para que sejam utilizadas por diferentes tipos de usuários.

A classificação convencional dos tipos de usuários de BD é apresentada na figura 2.2; segue uma descrição de [CHU83].

- (a) administrador de banco de dados (ABD) - utiliza as linguagens de definição de dados (LDD) e de manipulação de dados (LMD) e os utilitários do SGBD.
- (b) programador de aplicações (PA) - utiliza linguagens com procedimentos do tipo COBOL, PL/1, etc.
- (c) usuário interativo (UI) - utiliza programas interativos de consultas "LINGUAGENS DE QUARTA GERAÇÃO".
- (d) usuário paramétrico (UP) - utiliza programas de consultas com telas programadas.

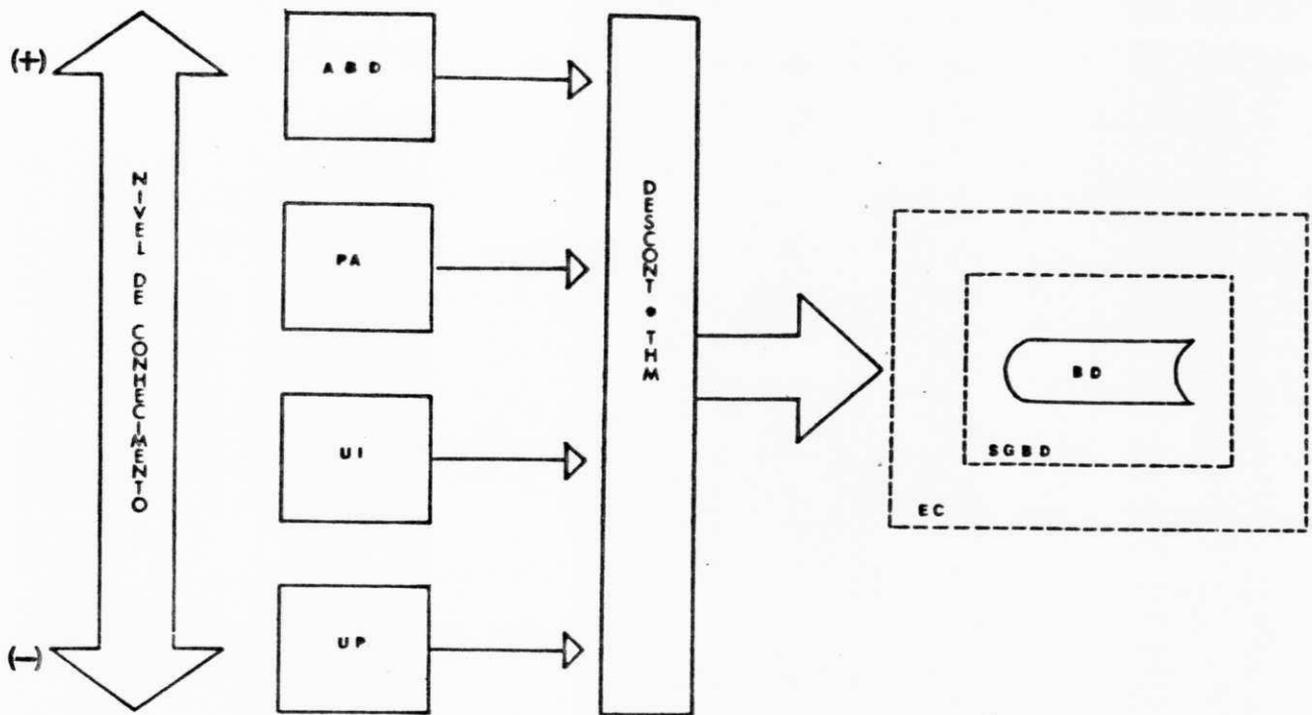


Figura 2.2 : Tipos de usuários de um BD.

2.3 Benefícios resultantes de um projeto ideal.

As disponibilidades técnicas que foram apresentadas neste capítulo objetivam, exclusivamente, ajudar um projetista de BD a amenizar a difícil tarefa exigida durante as fases de desenvolvimento de BD, e contribuir para uma análise mais meticulosa, tornando a riqueza semântica de uma modelagem cada vez mais completa. O resultado direto será o aumento da qualidade de um projeto de BD e a redução de seu ciclo de desenvolvimento.

Os parâmetros que influenciam na melhora da qualidade de um projeto são dependentes de :

- (a) análise detalhada das especificações de uma aplicação;
- (b) detecção de erros;
- (c) inconsistências;
- (d) omissões de informações;
- (e) identificação de redundâncias;
- (f) identificação de alternativas estruturais de um projeto;
- (g) análise de desempenho e estudos extensivos convenientes;
- (h) parâmetro adicional a ser considerado, que é a não modelagem de associações transitivas.

A redução do ciclo de vida de desenvolvimento de um projeto de BD, dá-se pela diminuição do número de interações e do tempo de cada interação.

3. MODELO DE DADOS: UMA TAXONOMIA.

Três gerações de modelos de dados estão amplamente difundidas; entretanto as pesquisas atuais direcionam-se no sentido de incorporar nova geração. Em [BROD84] é feita uma abordagem que engloba os seguintes modelos:

- (a) modelos de dados primitivos;
- (b) modelos de dados clássicos;
- (c) modelos de dados semânticos;
- (d) modelos de dados semânticos de propósitos especiais.

3.1 Modelos de dados primitivos.

Nos modelo de dados primitivos, a representação dos objetos está disposta em forma de registros, que por sua vez, são agrupados em arquivos. Os relacionamentos entre objetos são estabelecidos utilizando-se índices e listas invertidas, e as operações disponíveis são restritas, permitindo, apenas, leitura e gravação sobre os elementos dos arquivos.

3.2 Modelos de dados clássicos.

São identificados como modelos clássicos: o modelo de dados hierárquico, que é uma extensão dos modelos primitivos, o modelo de dados de rede, que abrange os conceitos do modelo hierárquico, e o modelo de dados relacional, que foi desenvolvido, a partir dos conceitos dos modelos hierárquico e de rede.

No modelo hierárquico, as denominações de arquivos, registros e itens (modelos primitivos), podem ser utilizados e permitem que registros se intitulem de segmentos e os itens, de campos. Isto porque no modelo hierárquico a terminologia não está padronizada.

Um BD que adote o modelo hierárquico será visto como sendo um conjunto de árvores, devido à representação física dos registros: um registro é dito ascendente e outro, descendente. Uma consideração de restrição é dada aos registros ascendentes, onde os mesmos podem ter vários descendentes, mas somente um ascendente [CHU83,BROD84].

O modelo de rede tem como elementos de dados a ocorrência de registros e o conjunto dessas ocorrências do mesmo tipo, caracterizando os tipos de registros. A formação da estrutura em rede ocorre quando o relacionamento de um conjunto de tipos de registros é feita através de ligações (referências especiais) e são implementadas sob a forma de ponteiros e lista de ponteiros.

O modelo relacional [CODD70] foi baseado no conceito matemático de uma relação. Neste modelo, uma relação é caracterizada por um conjunto de n-tuplas, onde uma tupla pode ser usada para representar os objetos e um relacionamento n-ário.

A versatilidade que o modelo comporta permite maior flexibilidade ao detalhamento sobre técnicas de implementações, e isola dentre os varios niveis de descrição de um BD, um nivel

abstrato compatível com uma formalização satisfatória e simples.

3.3 Modelos de dados semânticos.

A busca constante ao desenvolvimento de novos modelos de dados objetiva, principalmente, aumentar o poder de representação do mundo real.

Tais modelos propõem-se a tornar os conceitos mais entendíveis, a fim de representarem detalhes com mais clareza, quando modelados sob abordagens de um modelo clássico. A isto se denominou de modelos de dados semânticos, classificados nos seguintes grupos [BROD84]:

- (a) extensões diretas dos modelos clássicos;
- (b) modelos de dados matemáticos;
- (c) modelos de dados indecomponíveis;
- (d) modelos semânticos hierárquicos estáticos;
- (e) modelos semânticos hierárquicos dinâmicos.

3.3.1 Extensões diretas dos modelos clássicos.

O modelo estrutural [WE79] não faz distinção entre os objetos e o relacionamento entre eles e um exemplo direto são os esquemas relacionais.

O modelo objeto-papel [BACH77] foi baseado no modelo de rede com a noção de "papel". Um "objeto" pode participar de muitos

"papéis" em uma aplicação e pode ter diferentes propriedades para cada "papel". (Por exemplo: o "objeto" professor e "papel" chefe-de-departamento). Este modelo reduz redundâncias, por considerar cada "papel" como "objeto" separado e permitir a modelagem um número maior de aplicações orientadas.

O modelo entidade-relacionamento (E-R) [CHEN76] combina as características dos modelos de rede e relacional e faz distinção clara entre objetos e relacionamentos, representando-os como redes, nas quais os objetos (entidades) são nós e relacionamentos são as ligações entre os objetos.

A popularidade da utilização do modelo E-R para os projetos de BD de alto nível, deve-se à economia de conceitos e à difusão prévia das entidades e relacionamentos como modelagem natural de conceitos. O modelo E-R difere significativamente dos modelos clássicos do qual foi baseado.

3.3.2 Modelos matemáticos.

Grande parte da escolha pelo modelo de dados relacional é atribuída à base formal matemática (teoria dos conjuntos e cálculo dos predicados). Pesquisadores têm usado estas bases para definir formalmente e estender o modelo relacional. Três grupos são definidos como modelos predominantes: grupo baseado na teoria dos conjuntos [CHIL77], os baseados na lógica de primeira ordem [JACO82] e o baseado no instanciamento universal [ULLM80].

Esses modelos mantêm a notação formal e as definições para conceitos apresentados em outros modelos e, mais significativamente, mantêm ferramentas formais na análise das propriedades de aplicações e na resolução de problemas teóricos relatados.

Existe diferença entre a visão tradicional de um BD e os BD's que usam modelos que são baseados na lógica de primeira ordem. Na visão tradicional, os objetos são representados em um BD como instâncias de tipos de objetos definido no esquema correspondente. Tipicamente existe grande número de instâncias para cada tipo de objeto. Na visão lógica, objetos são representados como termos e propriedades (atributos, relacionamentos, restrições) como sentenças lógicas.

3.3.3 Modelos indecomponíveis.

Os modelos indecomponíveis [BROD84] foram inspirados na necessidade da representação de informações como fatos atômicos, ao contrário de grupos complexos de fatos. Nesses modelos, os fatos atômicos não podem ser decompostos em fatos separados.

O modelo de relacionamento-binário [BPP76] é também um modelo de dados indecomponível e corresponde à restrição do modelo relacional, no qual os relacionamentos são binários, em vez de n-ário. Um relacionamento binário representa relacionamento entre um objeto e um simples atributo, representando, assim, fato atômico. Um relacionamento binário é o

menor fato representável no modelo relacional.

No modelo relacional indecomponível [BROD84] as relações não precisam ser binárias. Os aspectos das propriedades obrigam à não decomposição das relações, significando que as informações serão perdidas, se uma relação for decomposta dentro de duas ou mais relações.

O modelo de dados mais largamente usado é o funcional que possui as características do modelo relacional, com a programação funcional. Os objetos são representados diretamente com suas propriedades e relacionamento representado como mapeamento funcional.

3.3.4 Modelo semântico hierárquico estático.

Muitos modelos de dados semânticos têm sido influenciados pela rede semântica, utilizada em inteligência artificial (IA). São integrados os conceitos do modelo de dados relacional com quatro importantes tipos de relacionamentos originais das redes semânticas que são: classificação, agregação, generalização e associação.

A classificação é uma forma simples de abstração de dados no qual um tipo de objeto é definido como um conjunto de identificações.

A agregação é uma forma de abstração onde um relacionamento entre objetos é considerado como o mais alto nível dos objetos agregados.

A **generalização** é uma forma de abstração que é similar e os objetos são também relatados para alto nível (objetos genéricos).

A **associação** é uma forma de abstração na qual o relacionamento entre objetos similares são considerados como conjunto de objetos do mais alto nível.

3.3.5 Modelo semântico hierárquico dinâmico.

É a extensão do modelo semântico hierárquico estático, e integra conceitos da modelagem dinâmica. Esse modelo integra, também, um conceito de linguagem de programação com conceitos de BD e faz uso de conceitos de tipos de dados avançados tais como: tipos de dados abstratos, classe e tipos multiformes.

3.4 Modelos de dados semânticos de propósitos especiais.

A quarta geração de modelo de dados surge para se dedicar a aplicações de propósitos especiais.

As aplicações particulares passaram a utilizar a teoria da modelagem semântica. Modelos anteriores têm assegurado conceitos de propósitos gerais para modelagem de muitos BD, assim como as linguagens de programação de propósitos gerais mantêm conceitos gerais para a programação. O desejo de se ter uma linguagem e modelos de altos níveis tem dirigido o desenvolvimento de um modelo de dados por aplicações específicas, tais como

procedimentos de escritórios, VLSI, CAD/CAM etc, assim como para aplicações tradicionais como estoque, contabilidade, folha de pagamento, etc.

Escolher um modelo de dados para uma aplicação particular é tão subjetivo quanto escolher uma linguagem de programação, um esquema de representação de conhecimento ou uma notação matemática. Quando os modelos de dados clássicos foram introduzidos, houve grandes debates sobre o relativo mérito dos mesmos. O que se nota agora é que os modelos não diferenciam significativamente em suas flexibilidades expressivas.

4. O MODELO SEMÂNTICO DE DADOS TEMPORAL HIERÁRQUICO.

4.1 Introdução.

O modelo de dados " Temporal Hierárquico " amplamente descrito em [CS85,HSY85,SCHI82,SCHI84] é um modelo semântico de dados que, sob o ponto de vista da abordagem de relacionamentos, baseou-se no modelo de relacionamento binário [BPP76], conforme descrito na seção 3.3.3.

O modelo THM apresenta, também, como características, três tipos de abstrações: generalização, agregação e agrupamento; considera os aspectos temporais e um tipo especial de relacionamento entre classes, denominado de "pré-pós"; oferece, ainda, facilidades para a modelagem dos aspectos estáticos (Esquema Conceitual de Dados - ECD) e dos aspectos dinâmicos (Esquema Conceitual de Operações - ECO) dos sistemas de informação.

O modelo semântico de dados THM dispõe de linguagem de especificação conceitual (LEC/THM) que é subdividida nas linguagens de definição de dados (LDD/THM) e de manipulação de dados (LMD/THM). A sintaxe da LEC/THM é apresentada nos apêndices A e B.

4.2 Taxonomia do modelo temporal hierárquico (THM).

A taxonomia representa a hierarquia dos conceitos do modelo semântico de dados THM (Fig. 4.1).

Conforme sua estruturação hierárquica é que serão realizadas consistências durante a modelagem de uma aplicação, com o objetivo de manter toda a riqueza semântica que o modelo THM comporta.

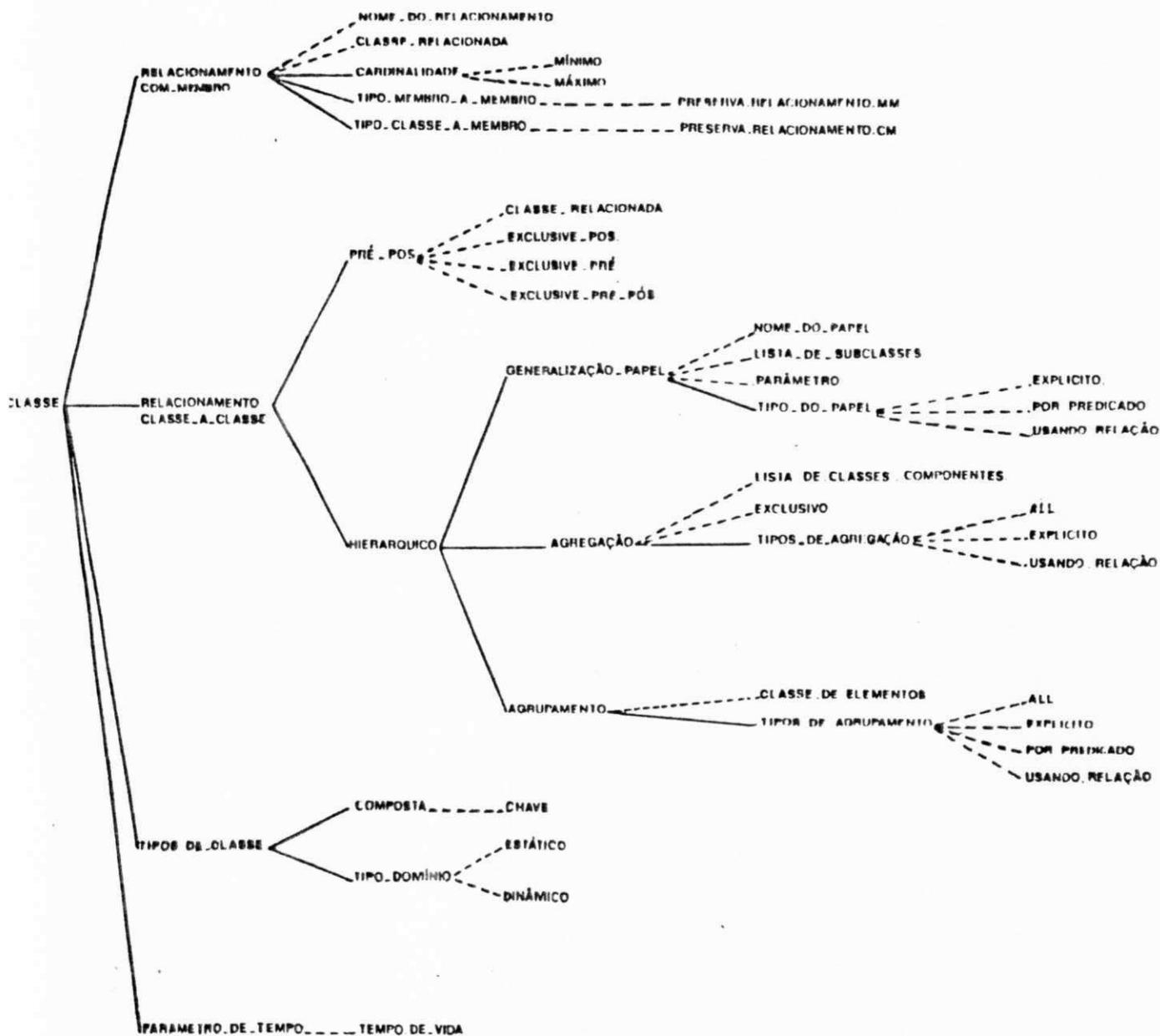


Figura 4.1 : Taxonomia do modelo THM.

4.3 Conceitos básicos.

Os objetos do mundo real (MR), que são de interesse de uma determinada aplicação, são representados como entidades ou membros de uma classe.

A fase de modelagem de um sistema de informação inicia-se pela formalização das estruturas de dados, também chamada de esquema conceitual de dados. Nesta fase, o modelo THM trata as entidades como sendo membros de classes e de relacionamentos; os membros de uma classe têm o mesmo espectro dos relacionamentos.

Os relacionamentos são classificados em três tipos: relacionamentos " membro-a-membro ", " classe-a-membro " e " classe-a-classe ". A figura 4.2 exemplifica esses tipos de relacionamentos.

(a) relacionamento membro-a-membro - caracteriza relação entre membros de duas classes (Fig. 4.2 Área I).

(b) relacionamento classe-a-membro - caracteriza relação entre uma classe e um membro de outra classe (Fig. 4.2 Área II).

(c) relacionamento classe-a-classe - caracteriza relação entre duas classes (Fig. 4.2 Área III).

Os tipos de relacionamentos estão inseridos em dois grupos de relacionamentos existentes, chamados de relacionamentos de

membros (quando uma entidade pertencente a uma classe destino, da qual o relacionamento faz parte, é um membro) e relacionamentos de classe (quando uma entidade pertencente a uma classe destino, da qual o relacionamento faz parte, é uma classe).

Para cada relacionamento é associada uma cardinalidade, representada por um par (MIN,MAX), significando que cada elemento (membro da classe origem) está associado a no mínimo ("MIN") e no máximo ("MAX") elementos (membros de uma classe destino ou classe relacionada).

Classe é um sistema composto de duas partes:

- (a) descrição da classe - composta do nome da classe, uma lista de relações com outras classes, informações sobre uma posição na hierarquia e parâmetro de tempo;
- (b) conjunto de membros - composto de um conjunto de entidades com a mesma abrangência dos relacionamentos.

As classes são distinguidas em dois tipos: classe composta e classe de domínio.

Na classe composta, os membros são identificados pelos relacionamentos com os membros de outra classe: por exemplo, as entidades da classe " MODELO " são identificados, através do relacionamento " tem-nome ", com a classe " NOME-MODELO " (Fig. 4.2 Area I).

A classe de domínio é um tipo de classe especial, cujos membros são caracteres inteiros ou outra estrutura básica, e seus membros são representados por sua própria identificação. Por exemplo, entidades da classe " NOME-MODELO " se auto-identificam (Fig. 4.2 Área I).

A classe de domínio se subdivide em classe de domínio dinâmica (onde seus membros têm representação explícita e podem ser modificados) e classe de domínio estática (onde seus membros têm representação explícita).

4.4 Conceitos dos relacionamentos hierárquicos entre classes.

Os relacionamentos hierárquicos são do tipo " relacionamento classe-a-classe ", identificados por " é-um " que representa uma generalização, " é-parte " que representa uma agregação, " é-elemento " para representar o agrupamento e um relacionamento temporal identificados por " pré-pós ".

Esses três primeiros conceitos são abstrações. Portanto, alguns detalhes da composição de entidades são suprimidos e entidades mais gerais são formadas.

Na generalização aplica-se um papel a uma classe para integrar a classe a uma lista de subclasses (Fig. 4.2 Área IV).

Na agregação, os membros (entidades) de uma classe podem ser formados pela fragmentação de entidades de classes diferentes (Fig. 4.2 Área V).

No agrupamento, os membros de uma classe podem ser agrupados, e esses grupos são tratados como entidade de uma classe de nível superior (Fig. 4.2 Área VI).

4.5 Modelagem do tempo.

No esquema conceitual de dados (modelagem estática) o tempo é modelado através de relacionamentos temporais " pré-pós ", relacionamentos com valores anteriores e de classe, com tempo.

- (a) relacionamento " pré-pós " - em muitas aplicações acontece que entidades removidas de uma classe são subsequentemente inseridas em outras classes do esquema modelado e, contrariamente, a entidade inserida nessa classe vem da primeira.

O relacionamento " pré-pós " é representado como $C1 \text{ >----> } C2$, e, no caso de mudança de conotação optativa para uma obrigatória, ou seja, de " >---> " para " >---->> ", haverá relacionamento " pré-pós-exclusivo " $C1 \text{ >---->> } C2$. Significa que toda entidade que deixa $C1$ deve obrigatoriamente passar para $C2$, e toda entidade nova de $C2$ provém da classe $C1$ (Fig. 4.2 Área VIII).

- (b) relacionamento com valores anteriores - se um valor deve ser mantido, utiliza-se o parâmetro " with old value " para garantir a preservação do valor em

operação de atualização. Essa garantia é assegurada quando é informado o período de existência do relacionamento pelos atributos " DE " e " ATÉ ".

- (c) classes com tempo - o parâmetro " with time " utilizado na modelagem de uma classe, considera a classe original como agregação com a classe " TIME-INTERVAL " que é responsável pela especificação do tempo de existência de uma entidade na classe original através dos atributos " DE " e " ATÉ ".

É apresentado na figura 4.2, exemplo completo da representação estrutural do esquema conceitual de um " Sistema de controle de veículos em um departamento de trânsito", exemplo sugerido por [GRI82].

A notação usada para a representação gráfica do esquema conceitual de dados é mostrado no apêndice C.

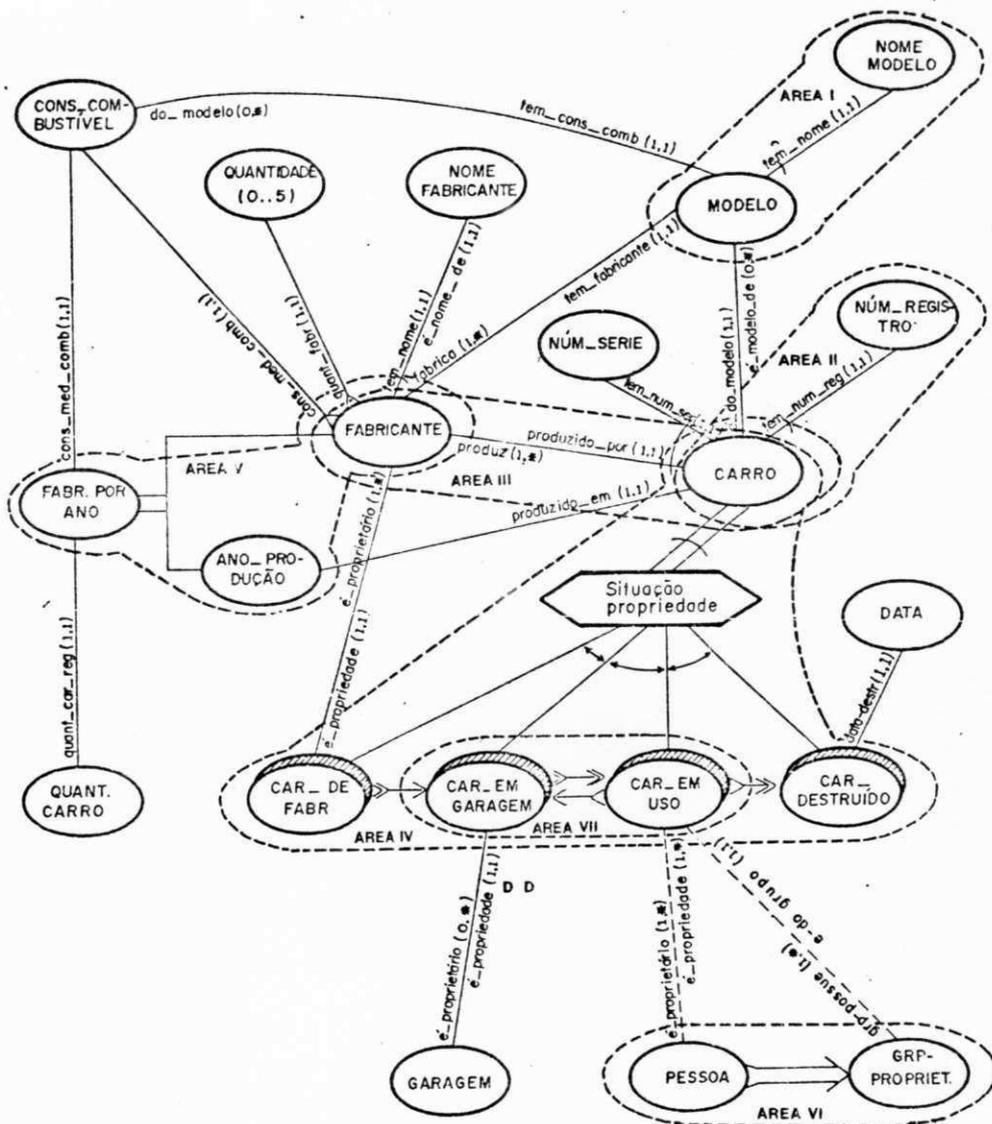


Figura 4.2 : Representação gráfica do esquema conceitual de dados de um sistema de controle de veículo em um departamento de trânsito.

4.6 Aspectos dinâmicos.

Esses aspectos serão tratados fartamente no sistema conceitual de operações THM (SISDECO-THM) que se encontra em fase de desenvolvimento.

Os aspectos dinâmicos são operações que manipularão os dados (aspectos estáticos) de uma aplicação. Na modelagem das operações, o THM dispõe dos conceitos de operações que são executadas quando pré-condições são satisfeitas. O THM também modela eventos e operações do tipo "triggers", que se encontram condicionadas às pré-condições dos eventos. Para garantir a integridade estrutural da modelagem conceitual, o modelo incorpora um sistema de eventos e "triggers" e um sistema de efeitos colaterais descritos em [TURA87, FERR87].

As operações primitivas (operações THM) são instruções que permitem inserções, remoções ou movimentações das entidades de uma classe e, estabelece, remove ou atualiza relacionamentos.

Os predicados primitivos são consultas que verificam pertinências das entidades em relação às classes. É mostrado aqui os predicados e operações primitivas tratadas no modelo.

Predicados primitivos:

(a) " e in C { at time } "

Determina se a entidade " e " pertence à classe " C " no intervalo de tempo (t).

(b) " e1 r e2 { at time } "

Determina se o relacionamento " r " está estabelecido entre as entidades " e1 " e " e2 " no intervalo de tempo (t).

(c) " is-part(e1,e2) "

Determina se a entidade " e1 " pertence à entidade agregada " e2 ".

(d) " is-elem(e1,e2) "

Determina se a entidade " e1 " é elemento da entidade de grupo " e2 ".

Os predicados primitivos não citados aqui, podem ser consultados em [FERR87] onde são feitas abordagens elucidativas quanto ao uso de predicados.

Operações primitivas:

(a) " insert e into C { at time } "

Insera a entidade " e " na classe " C " no tempo " t ".

(b) " delete e from C { at time } "

Remove a entidade " e " da classe " C " no tempo " t ".

(c) " establish e1 r e2 "

Estabelece relacionamento " r " entre as classes " e1 " e " e2 ". (Relacionamento de membros).

" establish C r e "

Estabelece relacionamento " r " entre a classe " C "

e uma entidade " e ". (Relacionamento classe a membro).

(d) " remove e1 r e2 "

Remove relacionamento " r " entre as entidades " e1 " e " e2 ".

" remove C r e "

Remove relacionamento " r " entre a classe " C " e a entidade " e ".

(e) " move e from C1 to C2 " (simplificação sintática).

delete " e " from " C1 "

insert " e " into " C2 "

Move a entidade " e " da classe " C1 " para a classe " C2 "; para isso, inicialmente remove a entidade " e " da classe " C1 " e, em seguida insere a entidade " e " na classe " C2 ".

(f) " update e1 r e2 to e3 " simplificação sintática).

remove e1 r e2 "

establish e1 r e3 "

Atualiza o relacionamento " r ", primeiramente removendo o relacionamento estabelecido entre as entidades " e1 " e " e2 " e, depois, estabelecendo um relacionamento entre as entidades " e1 " e " e3 ".

5. O SISTEMA DESCONT-THM.

Os avanços teóricos alcançados no desenvolvimento de esquemas conceituais (EC) de aplicações que utilizem BD, se por um lado trouxeram contribuições positivas em termos de abstração do MR, por outro apresentam alto grau de complexidade sob o ponto de vista de sua realização prática, dificultando seu tratamento por usuários de diversos níveis de capacitação.

É exigido dos usuários amplo domínio sobre metodologias e técnicas de desenvolvimento de aplicações, e as dificuldades em descrever formalmente o UD, por este ser nebuloso, não são atenuadas, e, com isso, não propiciam realização satisfatória.

As pesquisas direcionam-se no intuito de tentar solucionar, num aspecto mais abrangente, os problemas existentes nas fases de um projeto de BD.

Todavia, metodologias automatizadas que contribuem para o desenvolvimento de projetos de BD com uma qualidade elevada, estão voltadas para diferentes tipos de usuários [CERI85].

As ferramentas de apoio ao desenvolvimento de projetos de BD's consistem de programas que auxiliam na obtenção das informações informais, estruturando-as e especificando suas manipulações. Algumas ferramentas conhecidas para este fim são: NLDA, ISTDA, INCOD-DTE e DIALOGO, que são descritas em [CERI85].

O DESCONT-THM é uma das ferramentas essenciais que compõe a configuração geral do sistema THM, sendo responsável pela

captação semântica do UD e da modelagem dos aspectos estáticos ECD de uma aplicação para BD.

O sistema adotou, para representar o conhecimento sobre o modelo semântico de dados THM, a estrutura de "Frames" [FK85]. Torna-se possível, por conseguinte, realizar a modelagem de uma aplicação de maneira amigável, garantindo a riqueza semântica oferecida pelo modelo THM. Além de auxiliar na aprendizagem do modelo THM, através de um mecanismo de qualificação do conhecimento dos usuários, que foi incorporado ao sistema, permite maior diversidade de conhecimentos entre usuários e um ECD consistente.

5.1 Arquitetura do DESCONT-THM.

A figura 5.1 apresenta os módulos que compõem esta ferramenta e que são:

- (a) editor de "frames" - responsável pela inserção, alteração e remoção de "frames" do Banco de Conhecimento (BC);
- (b) modelagem do EC-THM - estabelece diálogo amigável com os usuários, através do qual capta a semântica do UD, armazenando-a nas tabelas de esquemas conceituais de dados (TECDs). A utilização do BC é feita através dos módulos de "Inferência e Explanação" para gerar uma TECD consistente;

- (c) banco de conhecimento (BC) - composto de um conjunto de "frames" estruturado de acordo com a taxonomia estabelecida pela metodologia THM, que implementa o conhecimento que o sistema utiliza;
- (d) módulo de explanação - dotado de um mecanismo de qualificação de conhecimento dos usuários (política de níveis) formalizando vários níveis de abordagens destinados aos usuários através do sistema de diálogo;
- (e) módulo de inferência - faz deduções dos procedimentos executados nos "frames" do BC durante a modelagem do EC-THM, tornando os dados captados consistentes e conclusivos;
- (f) TECDs - são as tabelas onde serão armazenadas as instâncias estabelecidas entre o sistema e os usuários. As tabelas comportam dados acerca de classes, relacionamentos, generalizações, papéis, agregações e agrupamentos (descritos no apêndice D).

5.2 Interface amigável.

O sistema DESCONT-THM é uma ferramenta destinada a aumentar a produtividade durante a modelagem dos aspectos estáticos e pode ser utilizada por usuários de diferentes níveis de conhecimentos.

Para que isso fosse possível, foi especificado uma "Interface Amigável" que é composta pelos módulos "Editor de Frames" e o "Sistema de Diálogo". O controle desses módulos é

feito através de telas programadas (menus) onde todas as funções podem ser acionadas via teclado (ver Fig. 5.1).

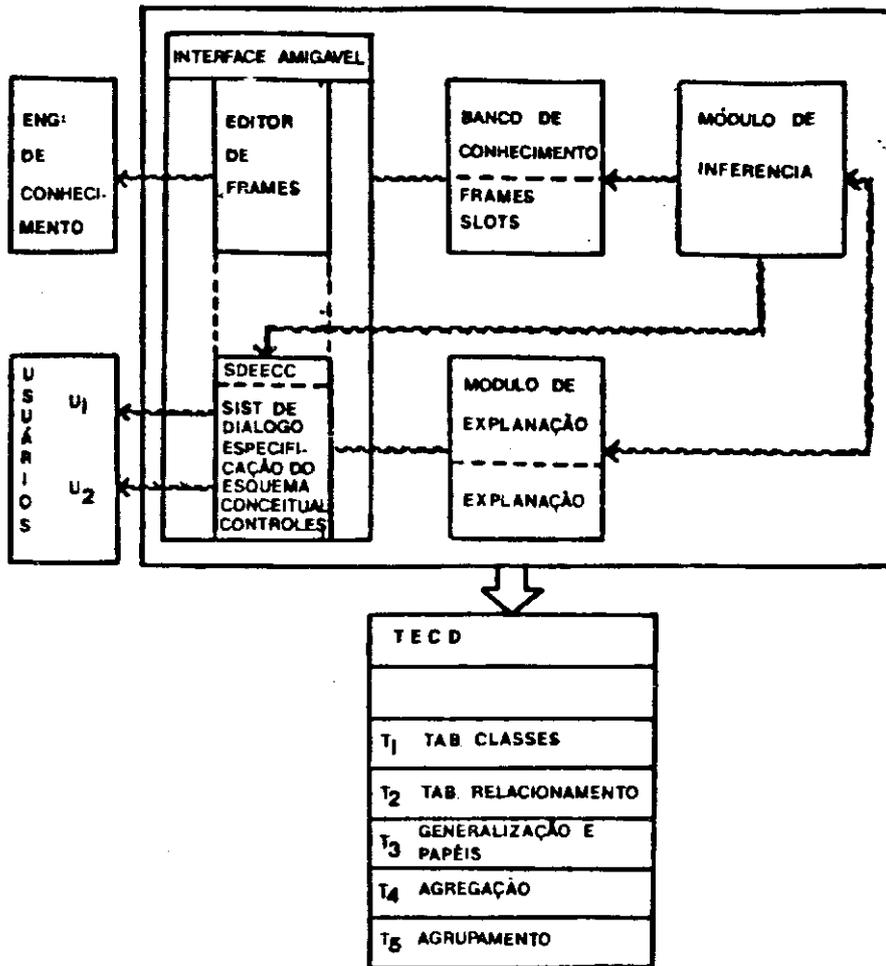


Figura 5.1 : Arquitetura do DESCONT-THM.

5.3 Representação do conhecimento (RC).

Para resolver problemas complexos, necessita-se essencialmente de conhecimento inerente a esses problemas e de um mecanismo que permita representar e manipular esse conhecimento.

O armazenamento de um conjunto de estruturas representativas

das relevâncias para resolver um problema concreto é especificado como BC [CHIA88].

Para implementar concretamente um BC é necessário contar com uma forma de RC. A RC é um conjunto de convenções sintáticas e semânticas que possibilitam descrever os objetos. Buscam-se nas RC's expressividade, poder heurístico e conveniência notacional [CT88].

No projeto de um BC devemos considerar a eficácia de uma das formas de RC e o grau de sucesso com que se consegue apresentar o comportamento dedutivo.

Das RC's comumente usadas em IA são apresentadas três classes fundamentais:

- (a) RC baseada em procedimentos : o conhecimento sobre o UD é expressado em sequência de procedimentos que indicam como devem ser manipulados tais conhecimentos [CHIA88];
- (b) RC baseada em regras : os chamados sistemas de produção são compostos de premissas e ações (que devem ser realizadas se forem cumpridas as premissas) ou de condições e conclusões que podem ser extraídas se cumpridas as condições [CT88];
- (c) RC baseada em objetos estruturados : nesta categoria encontram-se as "Redes Semânticas" e os "Frames". A

estas duas categorias será dado enfoque com amplitude significativa por ter sido este o tipo de RC adotado no DESCONT-THM. Ambos os modelos de RC têm em comum: os dados, onde o conhecimento está organizado perifericamente aos objetos e eventos de determinada aplicação; a estrutura básica de dados (por ser inspirada nos grafos) e as estruturas de registros.

5.3.1 Redes semânticas e frames: uma abordagem conceitual.

5.3.1.1 Redes semânticas.

As redes semânticas caracterizam-se pela conceituação de objetos (representados por nós) e relações entre eles (representados por arcos ligando os nós entre si e identificados como tipos individuais e genéricos) [CT88].

Os nós individuais representam descrições ou afirmações referentes ao valor individual de um objeto, e os nós genéricos referem-se a uma classe ou categoria de objetos.

Os conceitos (objetos) são dispostos em taxonomias e existem arcos especiais do tipo "é-um", "é-um-tipo-de". O arco "é-um" conecta um nó individual a um nó genérico e especifica uma classe qualquer. O arco "é-um-tipo-de" conecta dois nós genéricos entre si e especifica que em um conceito um tipo é subtipo de outro.

A estrutura de rede surge devido à particularidade existente nos nós genéricos que capacitam um nó genérico ser mais ou menos específico do que outro. O mesmo não ocorre com os nós individuais que têm caráter específico de mesmo nível, sendo extensivo a todos os nós.

Os arcos "é-um" e "é-um-tipo-de" permitem inferências devido a sua organização hierárquica em uma taxonomia. A interpretação dos arcos e nós pode ser ambígua, gerando com isso problemas de confiabilidade durante a realização de inferências.

Por exemplo:

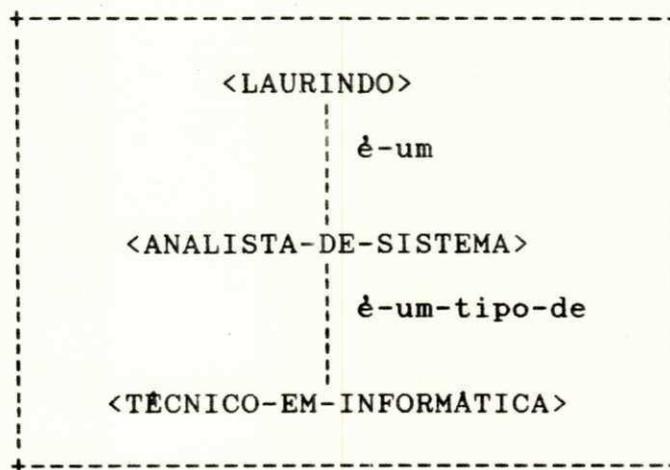


Figura 5.2 : Arcos especiais para conexão de nós.

5.3.1.2 Dedução nas redes semânticas.

Dois métodos de dedução aplicam-se às redes semânticas: inferência baseada na herança de propriedades e deduções por unificação de redes.

(a) herança de propriedade : Onde todas as propriedades dos nós superiores em uma hierarquia (obedecendo a uma taxonomia) são também pertencentes aos nós inferiores (Fig. 5.3).

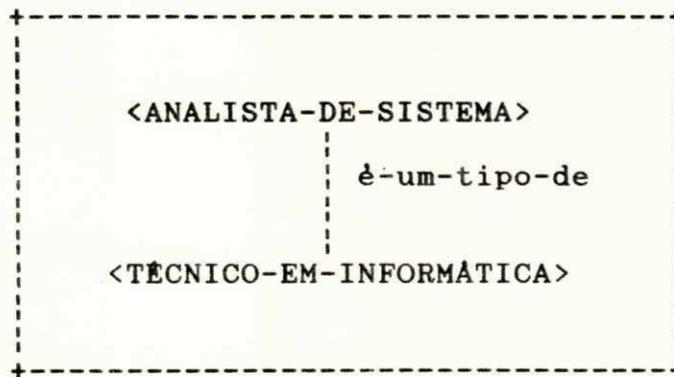


Figura 5.3 : Herança de propriedade.

Deseja-se incluir na rede um dado particular de que uma pessoa, "LAURINDO", é um ANALISTA-DE-SISTEMAS. Agrega-se, então, um nó e um arco à rede existente, modificando-a para a rede que é mostrada na figura 5.2, através da qual pode-se deduzir que "LAURINDO é-um TÉCNICO-EM-INFORMÁTICA".

A organização hierárquica permite, também, distribuir facilmente propriedades. Quando se deseja expressar que "Um TÉCNICO-EM-INFORMÁTICA faz CURSOS", complementa-se a rede conforme figura 5.4.

A herança de propriedades via arcos do tipo "é-tipo-de", além de representar economia cognitiva no sentido de eficiência no armazenamento, também serve para avaliar condições e manter restrições existentes, para que um objeto se encontre em uma dada posição dentro de uma taxonomia mais complexa.

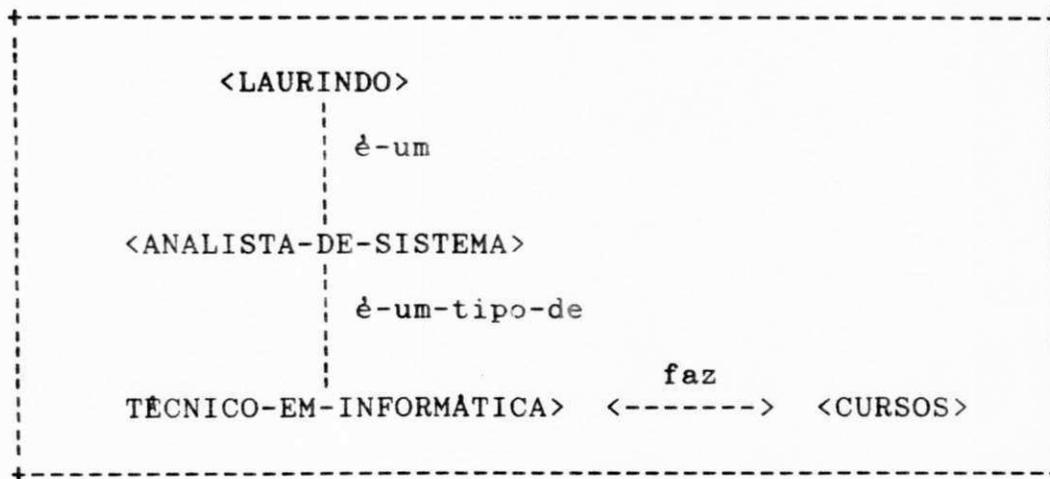


Figura 5.4 : Organização hierárquica, permitindo distribuição de propriedades.

- (b) dedução por unificação de redes : Este método necessita de uma rede pré-definida para que um fragmento externo possa ser unificado com o BC (conjunto de nós da rede pré-definida). O exemplo deste método de dedução é apresentado na figura 5.5.

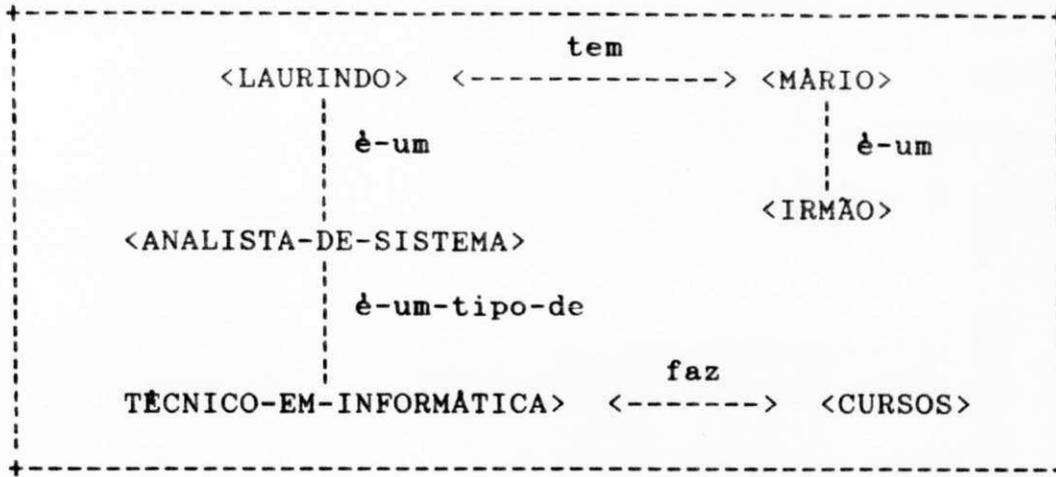


Figura 5.5 : Dedução por unificação

Quando for perguntado: " o que LAURINDO tem ? ", será construído um fragmento. Ao se comparar o fragmento com nó "MARIO" será atribuído este valor como resposta. (Fig. 5.6).

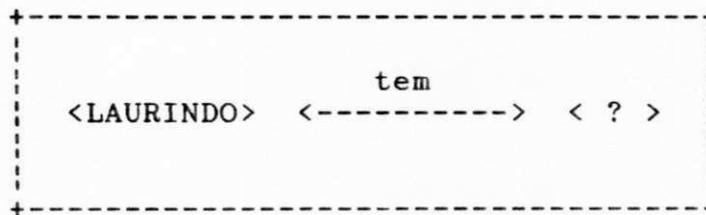


Figura 5.6 : Fragmento a ser unificado com o BC.

Para a pergunta: " existe algum TÉCNICO-EM-INFORMÁTICA que tem um IRMÃO ?". Utiliza-se a combinação da unificação com herança (Fig. 5.7).

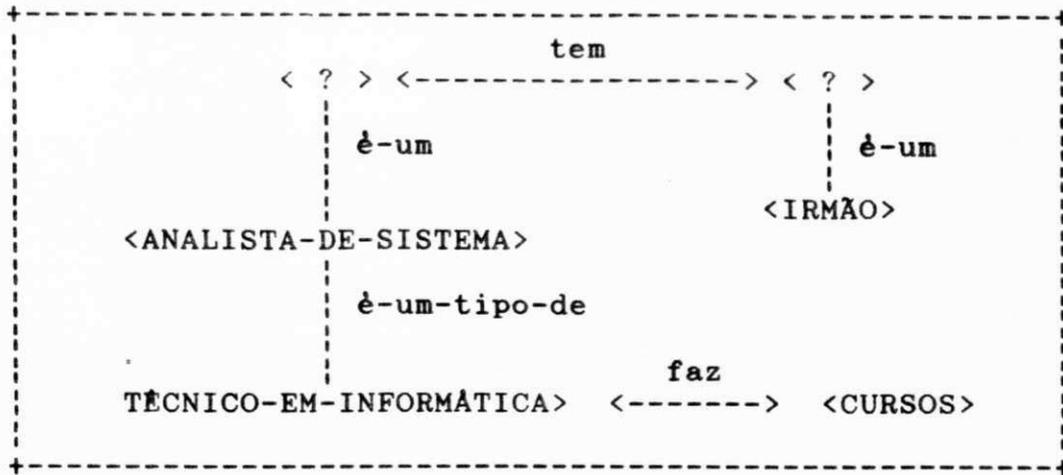


Figura 5.7 : Instanciamento como resultado da inferência submetida ao BC.

O conjunto de tipos de nós e arcos que compõem uma rede, subsidiará a qualificação da boa linguagem de rede.

Todavia, a falta de uma definição semântica uniforme, torna o formalismo da rede (facilidades de implementação) mais importante que o conhecimento instanciado aos nós e arcos.

5.3.1.3 Frames.

Os "frames" são evoluções das redes semânticas e são considerados nós complexos.

Essencialmente, um "frame" é uma descrição estruturada de um objeto ou classe de objetos [FK85] composta de "slots" que consistem de detalhes relevantes associados aos "frames". Tais

"slots", por sua vez, são constituídos de atributos, também conhecidos por "facetar".

A figura 5.8 apresenta o exemplo clássico de um sistema de "frames" descrito em [MINS75].

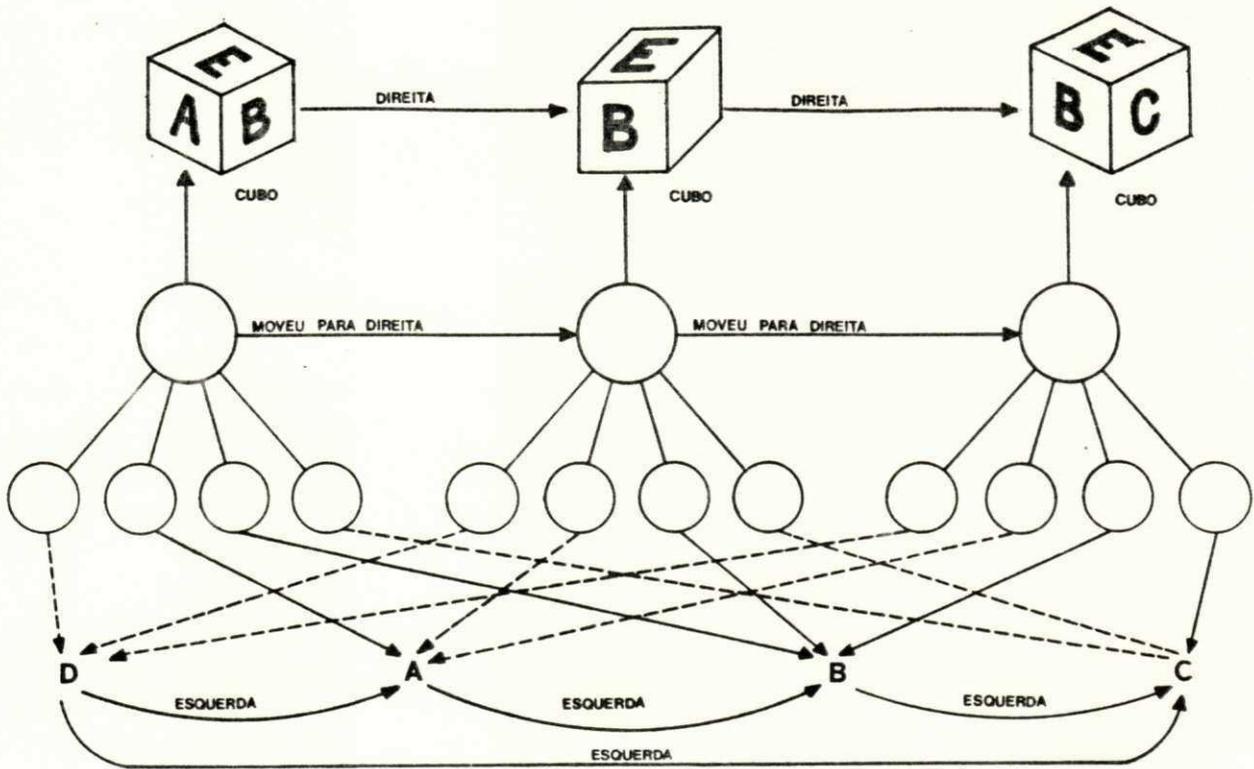


Figura 5.8 : Um sistema de frames.

Este sistema de "frames" consiste de três "frames", onde cada um representa uma perspectiva de um cubo.

As linhas cheias representam os atributos de uma perspectiva particular. As linhas tracejadas indicam faces que não são visíveis em relação à perspectiva particular. As ligações entre

os "frames" descrevem o relacionamento entre as visões que o "frame" representa.

Os "frames" são conectados através de compartilhamento de nós, onde os nós compartilhados representam a visão de uma face.

Cada "slot" de um "frame" pode estar interconectado a outra estrutura de "frame" e uma estrutura qualquer pode pertencer a mais de um "slot", resultando desta particularidade a não necessidade de armazenamento de informações redundantes.

Outros detalhes associados aos "slots" incluem:

- (a) valores - que são estabelecidos a um atributo por efeito, ou seja, via herança de "frames" de níveis superiores.

Por exemplo: Tratando-se do "frame" LADRILHO-1, um "subframe" do "frame" LADRILHO apresentado na figura 5.10 (1) herdará diretamente deste os "slots" "FORMA" e "COR", caracterizando, assim, inferência natural (Fig. 5.9).

- (b) restrições - que são estabelecidas às "facetas", envolvem um intervalo de valores definidos como valor mínimo e valor máximo.

(1) Este exemplo foi apresentado em [CT88].

Por exemplo: no "frame" LADRILHO, é atribuído um valor mínimo e outro máximo aos atributos do "slot" "COR", ou seja, "ESTOQUE-MÍNIMO" e "ESTOQUE-MÁXIMO", respectivamente (Fig. 5.10).

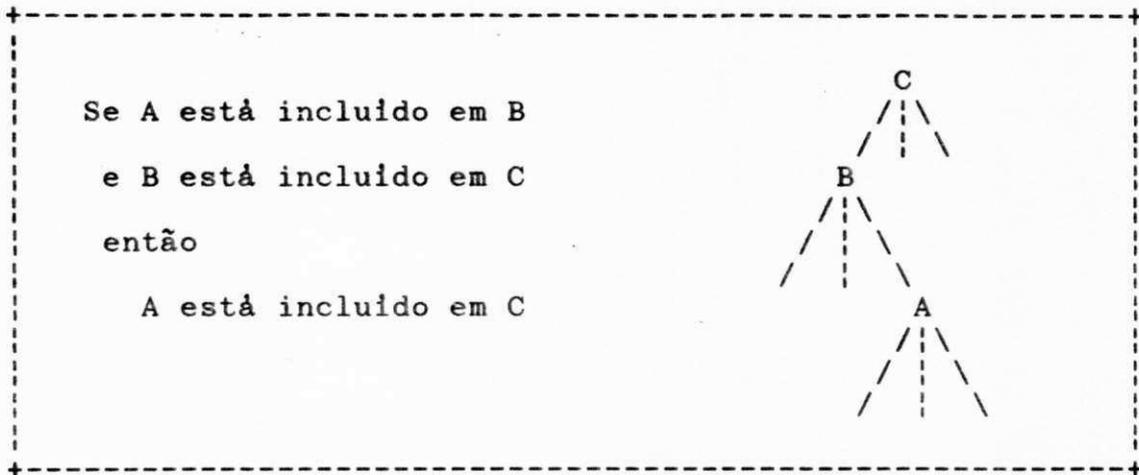


Figura 5.9 : Inferência natural.

- (c) **procedimentos agregados** - indicam como um atributo será instanciado por um procedimento existente em um "frame".

Por exemplo: no caso de se desejar calcular o "VALOR-MASSA" do "frame" "LADRILHO-1", o "SLOT-MASSA" pode chegar por herança de um procedimento armazenado no "frame" "BLOCO" do qual "LADRILHO" é-um-tipo que aplica a fórmula "MASSA = VOLUME * DENSIDADE". (Fig. 5.10).

Este tipo de procedimento está intrinsecamente ligado

ao estilo de programação orientado pelo objeto [CT88].

Foram associadas às noções de procedimentos orientados a objetos, com idéias contidas em [MINS75] (sobre estruturas de "frames"), à linguagem KRL[BW77] (linguagem baseada em "frames").

Um tipo de procedimento agregado, assemelhando-se aos modelos híbridos, é a integração de regras à técnica de "frames". Por isso, pode-se definir como sendo um "frame" especial, cujos "slots" correspondem às premissas e conclusões da regra.

Por exemplo: Na regra "Todo veículo com 6 ou mais rodas e mais de 5 toneladas de peso é um caminhão". Veja na figura 5.11 (2) um "frame" que expressa tal regra.

Portanto, um "frame" pode agrupar "Regras-de-Caminhão", constituindo ferramenta de inferência adicional.

A vantagem deste método de dedução (combinação) é que por meio de "frames" descritos de uma classe, estabelecem-se condições necessárias para que elementos lhe pertençam e as regras de produção ("slots" especiais) estipulem condições suficientes de pertinência; a verificação destas condições tornam uma inferência satisfatória.

(2) Este exemplo foi apresentado por [FK85].

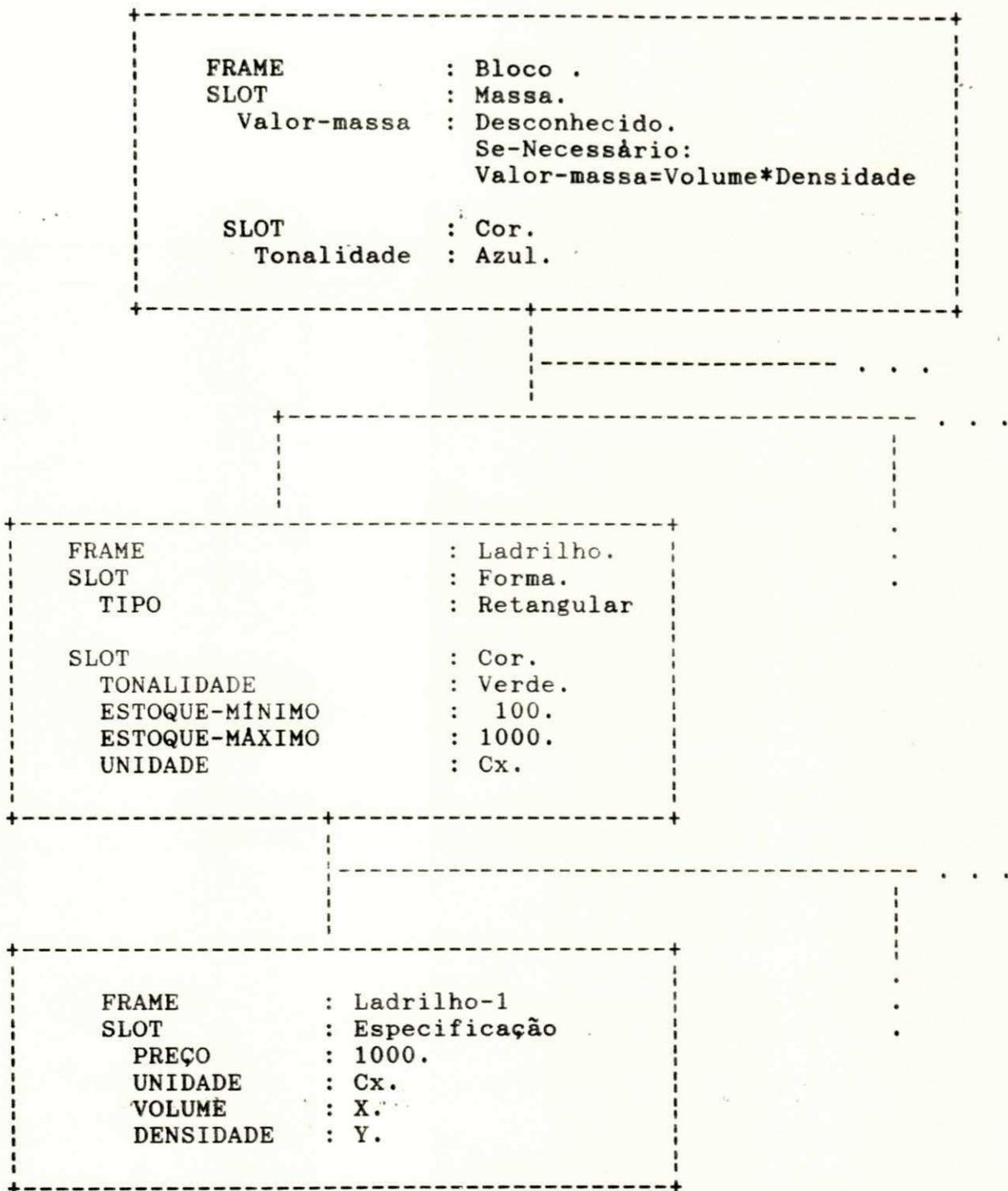


Figura 5.10 : Detalhes associados aos "frames" e "slots".

FRAME	: Regra-de-Caminhão.
PERTINÊNCIA	: Regra-de-Classificação-de-Caminhão.
SLOT	: Premissas.
TIPO-CONEXÃO	: Conjunção.
VALOR	: X é um caminhão.
	X tem Z rodas.
	X > = 6.
	X pesa Y toneladas.
	Y > 5.
SLOT	: Consequência.
VALOR	X é um caminhão

Figura 5.11 : Integração de regras à técnica de "frames".

5.3.2 A RC do modelo semântico de dados THM baseado em "frames".

Uma taxonomia para o modelo semântico de dados THM (abordado no capítulo 4) foi desenvolvida, formando a base para a RC do modelo THM. O objetivo desta taxonomia é estruturar os conceitos semânticos do THM sob a forma hierárquica.

Na taxonomia foi convencionado o uso de linhas cheias para caracterizar os níveis entre superclasses (superníveis) e subclasses (subníveis), e linhas tracejadas para explicitar a relação desses subníveis com os subníveis terminais.

É importante salientar que o padrão de referência da hierarquia é relativo, ou seja, um subnível será um supernível quando referenciado com um elemento do subnível. Por exemplo, o

conceito "RELACIONAMENTO-CLASSE-A-CLASSE" é supernível dos conceitos "PRÉ-PÓS" e "HIERARQUICO"; portanto ambos são subníveis.

Quando o referencial muda passando a ser o conceito "CLASSE-RELACIONADA", como elemento de "PRÉ-PÓS", o conceito "PRÉ-PÓS" torna-se supernível de "CLASSE-RELACIONADA".

Conforme a disposição hierárquica partiu-se para analisar o problema de como representar o conhecimento do modelo THM, e cada conceito é tratado como componente do BC, componente este representando elementos da semântica do THM com detalhes, valores, restrições e procedimentos agregados..

As características funcionais entre os "frames" e os conceitos do modelo THM (especificado e hierarquizado em uma taxonomia) tornou evidente e oportuna a utilização desta RC. Isso porque permitiu a representação do conhecimento do modelo THM, bem como a implementação na mesma estrutura hierárquica da taxonomia. Além disso, existem facilidades para manipulação do conhecimento, através de telas programadas embutidas em um editor de "frames", "EDFRAME", que executa um processo navegacional através da estrutura básica que é isomorfa à taxonomia.

Cada conceito da taxonomia do modelo THM corresponde a um "frame" respectivo, que obedece com rigor à semântica do THM para garantir a integridade semântica do BC.

Para que tal objetivo fosse atingido, utilizou-se um método

de diferenciação aplicado aos "slots" de cada "frame", inspirado na linguagem KEE [FK85] ilustrado nas figuras 5.12 e 5.13. Cada "slot" é especificado como "M_SLOT" se o "slot" pertencer a todos os "frames" de níveis inferiores, significando que todos os "subframes" a partir do "frame" que possui o "M-SLOT", herdarão este "slot", e "P-SLOT", caso o "slot" reflita propriedades exclusivas aos "frames" que a ele pertencem, não proporcionando herança, pois, não possui "subframes". Conclusivamente, os "frames" terminais só possuem "P-SLOT".

FRAME	: Classe.
SUPERNÍVEL	: Classe.
SUBNÍVEIS	: Relacionamento-com-Membros. Relacionamento-Classe-a Classe. Tipos-de-Classe.
M-SLOT	: M-Restrição-da-Classe.
FAC1	: Não-Númeroico.
FAC2	: Classe-Única.
FAC3	: Tamanho-Max-25.
FAC4	: Desconhecido. Se-Necessário: SN1 : Transforma-Maiúscula SN2 : Troca-Branco-Sublinhado
P-SLOT	: P-Variável-da-Classe.
FAC1	: Varclasse.

Figura 5.12 : Exemplo de distinção entre "slots".

FRAME	: Parâmetro-de-Tempo.
MEMBRO DE	: Classe.
P-SLOT	: P-Qte-de-Tempo-de-Vida.
FAC1	: Tempo-Localizado.
FAC2	: Valor.
FAC3	: Unidade.
P-SLOT	: P-Valor-Tempo-de-Vida.
FAC1	: [Ano/Mes/Dia/Hor/Min/Seg].
FAC-ANO-MIN	: 88.
FAC-ANO-MAX	: 99.
FAC-MES-MIN	: 1.
FAC-MES-MAX	: 12.
FAC-DIA-MIN	: 1.
FAC-DIA-MAX	: 31.
FAC-HOR-MIN	: 1.
FAC-HOR-MAX	: 24.
FAC-MIN-MIN	: 1.
FAC-MIN-MAX	: 60.
FAC-SEG-MIN	: 1.
FAC-SEG-MAX	: 60.

Figura 5.13 : Exemplo de integridade semântica do BC.

5.3.3 Descrição semântica do EDFRAME: O Editor de Frames do BC.

O EDFRAME auxilia o usuário a implementar e manter um BC com facilidades disponíveis no editor (inserir, eliminar e modificar os frames) permitindo que uma tarefa importante seja executada de maneira amigável.

Na figura 5.14 é apresentada a arquitetura do EDFRAME, que é parte da "INTERFACE AMIGÁVEL", composta dos módulos de "ESTRUTURA DE DADOS E CONTROLE", responsável pelo endereçamento e seleção dos "frames", e "CONTROLE DO ABD-THM" que trata da autorização de

acesso ao BC por usuários especialistas no modelo THM e da inicialização das estruturas básicas para o armazenamento do BC.

A função "MANUTENÇÃO DO BC" ativará o EDFRAME que utilizará a referência do processo navegacional, identificando o "frame" que se deseja editar.

A referência será fornecida pelo usuário, através da escolha de uma das telas programadas, que são executadas no processo navegacional, pela estrutura que foi carregada inicialmente (estrutura básica).

A figura 5.15 apresenta a estrutura carregada que serve de roteamento no processo para acessar um "frame". Nesta taxonomia de acesso, cada círculo representa um "frame" (nó terminal) e os quadrados são nós auxiliares (nós fechados) que levam a um outro nó auxiliar ou a um nó terminal. A numeração empregada a cada elemento, representa a referência para o acesso a um "frame".

Um exemplo ilustrativo do processo navegacional que ativa o EDFRAME e executa suas funções é mostrado no apêndice E.

Será editado um arquivo do tipo texto, cujo nome corresponde ao nó referenciado (acessado da estrutura básica ao EDFRAME com a extensão ".FRM").

Este editor é baseado em comandos, onde o usuário precisará utilizar as combinações das teclas de controle. A configuração padrão do editor usa uma sequência de teclas de comandos,

popularizadas pelo "WORDSTAR" (3), tornando, assim, mais fácil sua utilização.

A figura 5.16 exibe tabela com o conjunto de teclas de comandos que são funções do EDFRAME.

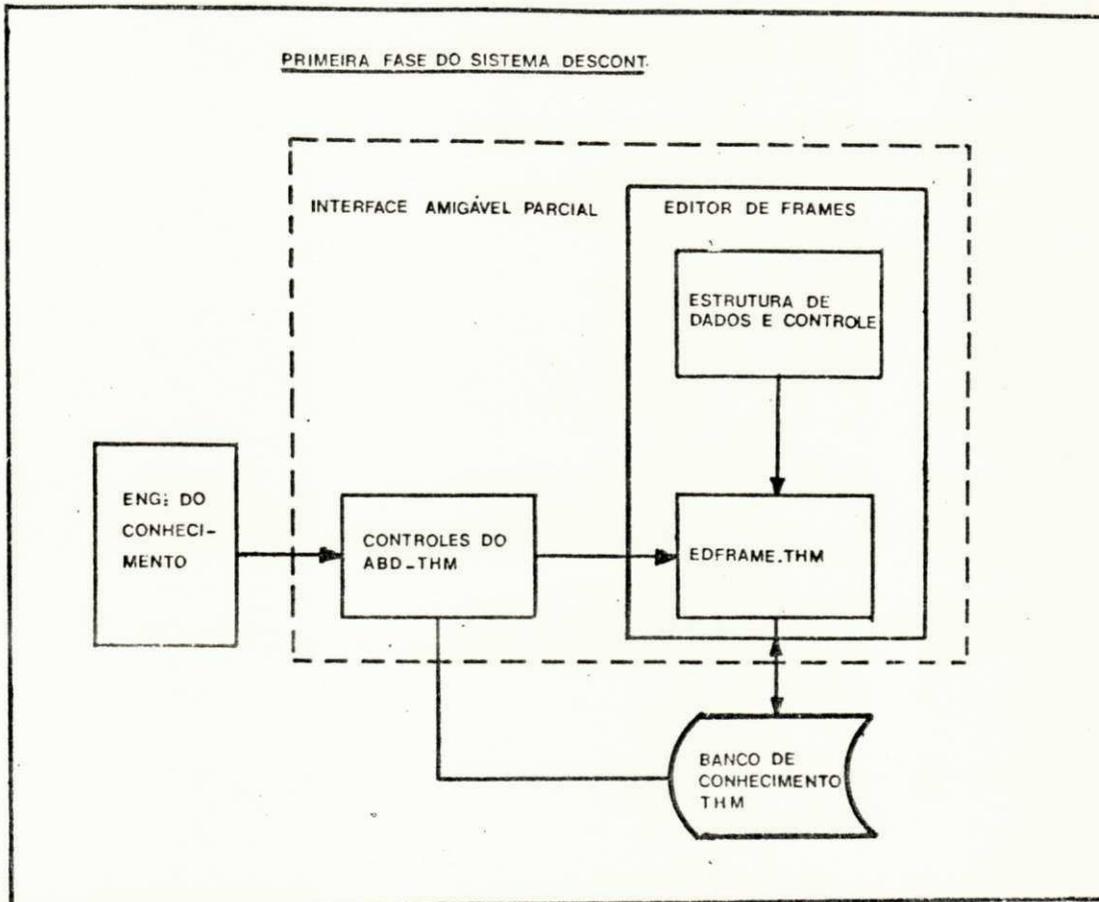


Figura 5.14 : Arquitetura do EDFRAME.

(3) Marca registrada da MicroPro International Corporation.

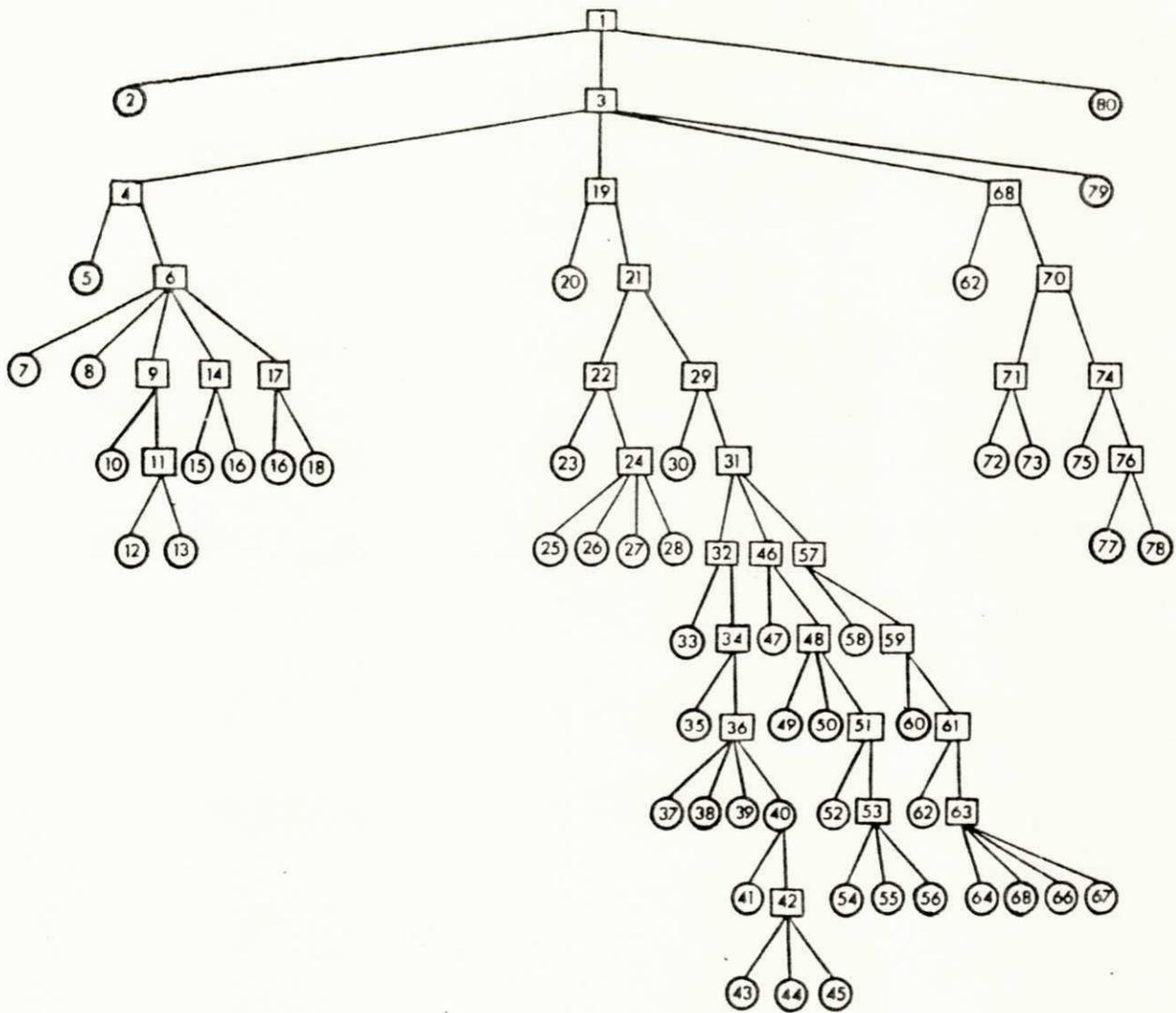


Figura 5.15 : Taxonomia de acesso aos "frames".

MOVIMENTO DO CURSOR	TECLAS
Move para caractere á esquerda	CTRL-S
Move para linha superior	CTRL-E
Move para linha inferior	CTRL-X
Move para palavra á direita	CTRL-F
Move para palavra á esquerda	CTRL-A
MOVIMENTAÇÃO DA TELA	TECLAS
Move tela uma linha para cima	CTRL-W
Move tela uma linha para baixo	CTRL-Z
INSERÇÃO	TECLAS
Modo de inserção	CTRL-V
Insera linha	CTRL-N
Insera linha se módulo de inserção ligado	<CR>
EXCLUSÃO	TECLAS
Elimina caractere sob o cursor	CTRL-G
Elimina palavra á direita do cursor	CTRL-T
Elimina linha	CTRL-Y
TABULAÇÃO	TECLAS
Move cursor para a próxima tabulação	CTRL-I
PARAGRAFAÇÃO	TECLA
Finaliza parágrafo	<CR>
GRAVAÇÃO DE ARQUIVOS	TECLAS
Grava e retorna ao menu de origem	CTRL-D

Figura 5.16 : Teclas de comandos do EDFRAME.

5.4 Módulo de explanação.

Por se tratar de um sistema que será manipulado por diferentes tipos de usuários, foi adotada uma política de hierarquização dos graus de conhecimentos, que os mesmos possam ter sobre sistemas de informação e BD. (função "MANUTENÇÃO DO EC-THM"). Da mesma forma o mecanismo de qualificação de conhecimento de usuários, é realizado através de procedimentos que estabelecem uma política de níveis.

Quando a primeira explanação for solicitada, um nível de seu conhecimento já estará determinado.

A explanação poderá ser solicitada quando ocorrer um erro durante a exibição de uma mensagem, ou no início de uma etapa da modelagem (onde são apresentadas as funções específicas da etapa).

Cada etapa da modelagem de uma aplicação corresponde a um nó na taxonomia do modelo THM e, conseqüentemente, a uma tabela de explanação. Todo nó referencia um índice que, por sua vez, identifica o elemento da tabela de acesso aos "frames", que possui no campo "arqhlp" o nome da tabela que será utilizada na explanação.

O nível de abordagem de uma explanação dependerá do nível de conhecimento que lhe foi atribuído anteriormente. Para melhor aproveitamento das explicações, é estabelecida uma política de níveis, onde o nível do usuário se modifica à medida

que seu grau de satisfação (quanto ao uso da explanação) também se modifica.

O algoritmo da política de níveis escrito em pseudo-código é mostrado no capítulo 6.

5.5 Módulo de inferência.

O conhecimento do modelo THM está representado sob forma de árvore, onde cada nó possui um conjunto de sucessores (nós filhos), que representam um posicionamento dos conceitos na taxonomia do modelo THM.

A estratégia de controle (formalismo computacional) adotada como procedimento de direcionamento da consistência, quando processados valores nos "frames", é o método "DEPTH-FIRST INFORMADO" (busca em profundidade com heurística local) [KVIT88], e com encadeamento "FORWARD" e "BACKWARD".

Durante a modelagem do EC-THM, o sistema utilizará uma rede de inferência, idêntica à estrutura hierárquica representada na taxonomia desenvolvida para o acesso aos "frames" (ver Fig. 5.15).

Nesta rede, os nós terminais indicam quais os conceitos que poderão ser instanciados durante a especificação de uma aplicação para qualquer classe identificadas no UD.

5.5.1 O Método Depth-First Informado.

Este é um método de busca de propósito geral. As técnicas de busca descrevem, independentemente, qualquer tarefa particular ou domínio de um problema [KVIT88]. A eficácia e eficiência do método depende da maneira como se exploram os conhecimentos que são aplicados a um problema particular.

Apesar da limitada efetividade dos métodos para resolver problemas complexos, são tidos como processos adequados, onde o conhecimento específico do domínio pode ser organizado e explorado, constituindo o núcleo de muitos sistemas de IA.

No método "depth-first", primeiramente expande-se a raiz, elegendo-se e expandindo-se a seguir o nó mais à esquerda, e ignora-se, momentaneamente, os demais nós. Segue-se, desta maneira, até o nó mais profundo, para se encontrar a solução (um nó terminal), ou até que não existam mais sucessores. Neste último caso, retrocede-se ("backtraking") até o nó mais recente que contenha um nó não explorado (nó não terminal). Na árvore (Fig. 5.15) os valores atribuídos a cada nó indicam a ordem em que os nós foram expandidos.

O método "depth-first" é muito utilizado por manter na memória os nós abertos ou ponteiros aos mesmos. Por exemplo, se a profundidade de uma árvore é P e o fator de ramificação R (número máximo de filhos), será necessário alocar $P \cdot R$ nós na memória, e se realizado "backtracking", somente será necessário alocar P nós ou P ponteiros.

É importante atentar para uma desvantagem neste método. Se na árvore anterior o nó terminal é o início para outra sub-árvore muito grande (podendo ser infinita e não conter solução), perder-se-ia tanto o esforço dispendido à procura da solução quanto seu retorno ao caminho válido; caso fosse infinito jamais poderia ocorrer esse retorno.

Na rede de inferência gerada pelo DESCONT-THM, isso não ocorre devido à taxonomia ser estática e cada nó terminal ser uma solução, não existindo caminhos infinitos.

O método compreende um processo exaustivo para encontrar caminhos aos nós objetivos devido à explosão combinatorial, pois expandem nós em demasia antes de encontrar o caminho desejado, tornando quase que impraticável sua utilização.

Para resolver este tipo de problema é sugerida a incorporação de mecanismos heurísticos que influenciarão na etapa de seleção das soluções.

Para o mecanismo heurístico adotado, a tabela conjunto de futuras soluções ("Tabfs") possui todas as possíveis soluções permitidas a uma classe, que são armazenadas em seus registros, onde cada registro identifica uma solução completa (conjunto de índices) que deverá ser instanciada à medida em que os conceitos forem sendo estabelecidos (resultado de processamentos dos "frames" de cada elemento desse conjunto).

Tais soluções influem notavelmente como informações sobre

qual alternativa é mais promissora com respeito á possibilidade de se modelar um índice (nó terminal).

Uma informação adicional acerca da aceitação do índice está armazenada na tabela de soluções eleitas ("Tabmeta"), onde estão as soluções já modeladas (índices aceitos).

Se o nó que foi atingido pelo processo navegacional (via método "depth-first") existir em "Tabfs" e "Tabmeta", será deduzido que tal conceito já foi modelado de forma completa (quando os registros de ambas as tabelas forem iguais) ou, caso contrário, indicará quais os conceitos que ainda faltam ser modelados (caso os registros das tabelas sejam parcialmente iguais).

5.5.2 Encadeamento "forward" e "backward".

Como o objetivo de um procedimento de busca é encontrar um caminho entre a configuração inicial (raiz) e a final (nó terminal), foi necessário designar duas direções para este processo.

- (a) "Forward" (para frente): Consiste na aplicação de operações integradas ao processo navegacional, através de telas programadas a partir do estado inicial e, seguidamente aos seus sucessores, até que alcance um estado final na rede de inferência, onde será iniciada a modelagem do referido conceito da aplicação com

processamentos das "facetas" dos P_SLOTS do "frame".

- (b) "Backward" (para trás): Consiste em aplicar operações a um dos estados finais recém - instanciados (após processamentos dos P_SLOTS) do "frame" (um nó terminal), então, retrocede-se ("backtracking") até o estado mais recente (nó pai) reiniciando um processamento de "frames" (desta feita nas "facetas" dos M_SLOTS) seguindo até que se atinja o nó inicial (raiz) da rede de inferência.

Os algoritmos que descrevem o módulo de inferência são descritos no próximo capítulo.

6. DESCRIÇÃO DA ESTRUTURA DE DADOS E EXEMPLO DE UMA MODELAGEM COM VISÕES DAS ESTRUTURAS INTERNAS DAS TABELAS.

O sistema DESCONT-THM foi implementado em TURBO PASCAL, para ser executado em ambiente de microcomputadores compatíveis com IBM PC. A escolha deveu-se, principalmente, à grande popularidade alcançada por esta linguagem, garantindo, por conseguinte, a portabilidade; além disso, pelo fato de a mesma ser dotada de módulo de compilação muito rápido, possibilitará a geração direta de um código executável.

Outras características importantes são da linguagem comportar programação estruturada, possuir bom desempenho tanto no desenvolvimento quanto na execução, além de prover grande número de comandos e de estruturas de dados, incorporando facilidades para gerar gráficos, boa legibilidade, e sendo de fácil aprendizado.

Entretanto, durante a fase de programação, exige sólida formação em estrutura de dados por parte do programador.

Outros sub-sistemas que compõem a configuração geral do THM foram ou estão sendo desenvolvidos em TURBO PASCAL.

6.1 A estrutura de dados do DESCONT-THM.

Os "frames" do BC e os dados instanciados pelo sistema de diálogo estão estruturados nas seguintes tabelas:

- (a) tabela de endereçamento de tarefas ("Tabrefas");

- (b) tabela de senha de usuários ("Tabsenha");
- (c) tabela de formatos ("Tabform");
- (d) tabela de conteúdo ("Tabcon");
- (e) tabela de acesso aos frames ("Tabaccess");
- (f) tabela de descrição ("Tabdescr");
- (g) tabelas de dados modelados ("Tabdat");
- (h) tabelas de frames ("Tabfrm");
- (i) tabelas de explicações ("Tabhlp");
- (j) tabela conjunto de futuras soluções ("Tabfs");
- (k) tabela de soluções eleitas ("Tabmeta");

6.1.1 Tabela de endereçamento de tarefas ("Tabrefas").

As funções do sistema DESCNT-THM estão armazenadas nesta tabela, conforme mostrado na figura 6.1.

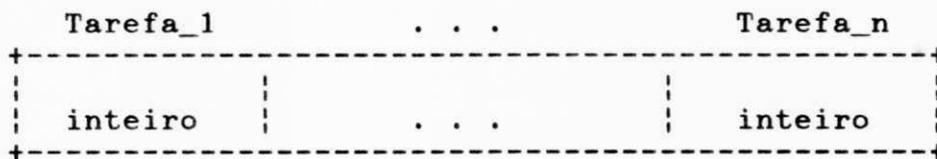


Figura 6.1 : Estrutura do registro de endereçamento de tarefas.

Cada campo do registro é um ponteiro para a "Tabform", com exceção do campo Tarefa_1, que é um ponteiro para o primeiro registro da "Tabsenha".

6.1.2 Tabela de senha de usuários ("Tabsenha").

Esta tabela é acessada quando o campo Tarefa_1 de "Tabrefas" for acionada. A mesma é utilizada para averiguar a existência de usuários que foram habilitados a executar a manutenção do BC-THM.

Cada entrada da "Tabsenha" é formada por 2 campos, conforme é apresentado na figura 6.2, sendo:

Nome - é o nome do usuário habilitado a executar a Tarefa_1 da "Tabrefas".

Senha - é o código de habilitação do usuário.

Nome	Senha
string[8]	string[8]

Figura 6.2 : Estrutura do registro da tabela de senha.

6.1.3 Tabela de formato ("Tabform").

A "Tabform" representa a estrutura principal do DESCNT-THM, onde estão armazenados todos os nós, correspondendo à taxonomia do modelo THM (Fig. 4.1).

Indice	Filho_1 . . . Filho_n	Pai	D_1...D_4	Ct_1	Ct_2
int.	int. . . int.	int	int ...int	int.	int

Figura 6.3 : Estrutura dos elementos da tabela de formato.

Cada entrada em "Tabform" representa um nó, composto pelos seguintes campos, conforme mostra a figura 6.3.

Índice - número associado ao nó, que é a representação simétrica de um conceito na taxonomia THM, durante o processo de acesso ao "frame" ou de instanciação de variáveis.

Filho_1 ... Filho_n - são ponteiros para os nós, filhos deste elemento.

Pai - é um ponteiro para o nó pai deste elemento.

D_1 ... D_4 - contém quatro valores para o dimensionamento da tela, que será apresentada quando for referenciado o registro em um acesso ao "frame".

Ct_1 - endereço do registro inicial da "Tabcon" que contém o início do texto para a tela de referência do elemento.

Ct_2 - endereço do registro final da "Tabcon" que contém o final do texto para a referência do elemento.

6.1.4 Tabela de conteúdo ("Tabcon").

Cada registro desta tabela (Fig. 6.4) é formada pelos campos:

Nro_reg - número do registro na "Tabcon" que indica o início ou final da descrição de um nó.

Conteúdo - linha com o texto de descrição para um nó referenciado.

Nro_reg	Conteúdo
inteiro	string[65]

Figura 6.4 : Estrutura dos registros da tabela de conteúdo.

6.1.5 Tabela de acesso aos frames ("Tabaccess").

Esta tabela possui os endereços de um conjunto de "frames" (supernível e subníveis) de um elemento da "Tabform" e possui, também, os nomes das tabelas de "frames", dados e explicações para o mesmo elemento que são detalhados na figura 6.5.

Indice	Spnível	Sbnível_1...Sbnível_2	Arqdat	Arqfrm	Arqhlp
inteiro	inteiro	inteiro ...inteiro	str[12]	str[12]	str[12]

Figura 6.5 : Estrutura da tabela de acesso aos frames.

Indice - número do registro de acesso que indica o nó determinado para o acesso a um "frame" e dados para o respectivo conceito do THM.

Spnivel - endereço de um registro da "Tabdescr" que identifica o supernível ("superframe") de um nó.

Sbnivel_1 ... Sbnivel_n - endereços dos registros da "Tabdescr" que identificam os subníveis (subframes) de um nó.

Arqdat - nome da tabela de dados para acessar ou armazenar valores da instanciação.

Arqfrm - nome da tabela de "frame" para acessar ou atualizar "frames".

Arqhlp - nome da tabela de explanação para acessar os conceitos ou exemplos do modelo THM.

6.1.6 Tabela de descrição ("Tabdescr").

Esta tabela contém registros com a descrição de todos os nós ("superframes" e "subframes") que fazem parte da estrutura principal ("Tabform").

Nro_descr	Descrição
inteiro	string[25]

Figura 6.6 : Estrutura da tabela de descrição.

A tabela de descrição é formada pelos campos:

Nro_descr - corresponde à identificação de um "superframe" ou "subframe" acessado.

Descrição - nome de um nível associado a um "frame".

6.1.7 Tabelas para dados modelados ("Tabdat").

A "Tabdat" é um conjunto de tabelas de mesmo tamanho que armazenam valores instanciados durante uma modelagem.

A "Tabaccess" referenciará a "Tabdat" desejada, representando um nó ou um conceito.

Cada elemento de uma "Tabdat" está estruturado conforme figura 6.7.

Indice	Status_m	Val_1	. . .	Val_n
inteiro	inteiro	inteiro	. . .	inteiro

Figura 6.7 : Estrutura de uma tabela para dados modelados.

Indice - número do nó (na taxonomia) a que esta tabela pertence.

Status_m - código que identifica se a modelagem de um conceito foi realizado com sucesso.

Val_1 ... Val_n - são valores instanciados durante a modelagem dos objetos.

6.1.8 Tabelas de "frames" ("Tabfrm").

A "Tabfrm" é um conjunto de tabelas com registros (tipo texto) que armazenam a descrição da semântica do THM (Fig. 6.8).

A tabela "Tabaccess" referencia um "frame" selecionado dentro da taxonomia para processar funções de consistência antes de aceitar os valores que serão armazenados na "Tabdat".

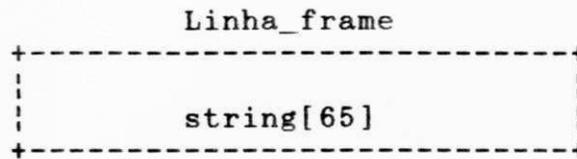


Figura 6.8 : Estrutura de uma tabela de frame.

O elemento da tabela é formado por um único campo:

Linha_frame - linha com texto de descrição do "frame".

6.1.9 Tabelas de explicações ("Tabhlp").

A tabela "Tabhlp" é um conjunto de tabelas com registros do tipo texto que armazenam informações (definições e exemplos) sobre o modelo THM, e são invocadas conforme necessidade do usuário ou quando o sistema exibir uma mensagem de erro.

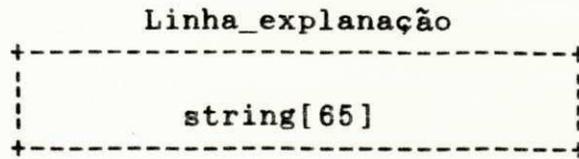


Figura 6.9 : Estrutura de uma tabela de explanação.

O elemento que forma a "Tabhlp" é:

Linha_explanação - linha com texto de descrição de conceitos ou exemplos.

6.1.10 Tabela conjunto de futuras soluções ("Tabfs").

Esta tabela contém, de forma sequencial por grupo, todas as soluções da modelagem que podem ser conseguidas por uma classe.

A tabela é de grande importância, pois avaliará as instanciações para a futura solução que será eleita para uma classe modelada (Fig. 6.10).

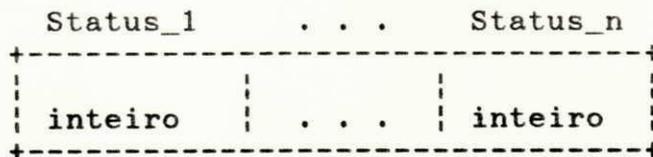


Figura 6.10 : Estrutura da tabela conjunto de soluções.

Status_1 ... Status_n - lista de números de nós que representam soluções eleitas

dentro de um grupo de registros da "Tabfs".

6.1.11 Tabela soluções eleitas ("Tabmeta").

Esta tabela possui a lista dos nós folhas da taxonomia do THM (Fig. 4.1) que foram modelados (um elemento da "Tabfs"), e servirá para acessar as tabelas de dados ("Tabdat") durante a geração das TECDS (Fig. 6.11).

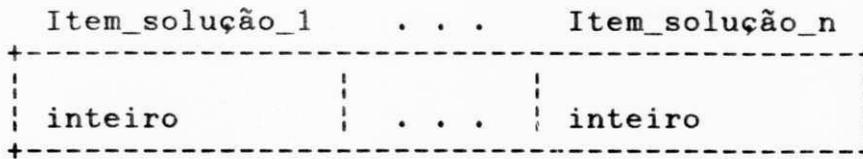


Figura 6.11 : Estrutura do registro da tabela de soluções eleitas.

Item_solução_1
.
.
Item_solução_n - valores dos nós modelados como solução.

6.2 Exemplo de uma modelagem e sua representação interna.

No intuito de acompanhar com maior clareza as etapas compreendidas numa modelagem, é mostrado o funcionamento das várias etapas em forma de algoritmo escrito em pseudo-código, utilizando a representação gráfica de um ECD sugerido por [CS85]

e apresentada na figura 6.12.

A exemplificação dar-se-á a partir da tela de menu principal com funções para modelagem do EC-THM. Os procedimentos anteriores a esta ação do sistema (instalação, geração das estruturas e manutenção do BC-THM) são descritos no tutorial do sistema DESCNT-THM apresentado no apêndice E.

A geração das TECDS é o resultado da interação do sistema com os grupos de usuários que o utilizarão (informando os aspectos semânticos de uma aplicação igual á mostrada na figura 6.12).

Um procedimento anterior ao de seleção da função para MODELAGEM DO EC-THM, (função geração das estruturas), armazenará as estruturas básicas na memória ("Tabsenha", "Tabcon", "Tabform", "Tabaccess", "Tabdesc", "Tabfs" e "Tabmeta").

As tabelas servirão de apoio tanto para a "MANUTENÇÃO DO BC-THM" quanto para a "MODELAGEM DO EC-THM, e são exemplificados nas figuras 6.13, 6.14, 6.15, 6.16, 6.17 e 6.18.

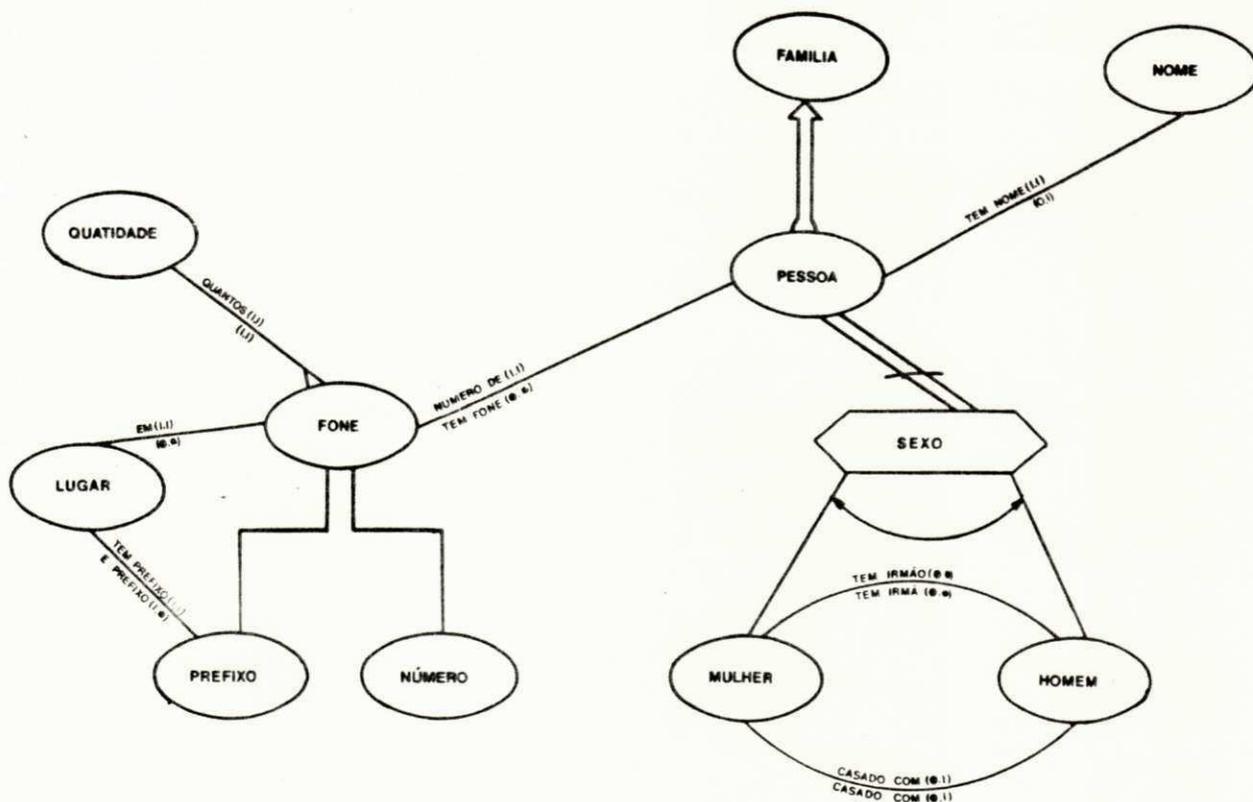


Figura 6.12 : Esquema conceitual de dados para um sistema telefônico simples.

Nome	Senha
LAURINDO	maogat
SLNV	nvbt
RLEE	itty9
.	.
.	.
.	.

Figura 6.13 : Organização dos dados na TABSENHA.

Nro_reg	Conteúdo
1	==> MENU PRINCIPAL <==
2	1 . CLASSE.
3	2 . SUB_FRAME DE CLASSE.
4	3 . SAIDA DO SISTEMA.
5	==> SUB_FRAME DE CLASSE.
.	.
.	.
.	.

Figura 6.14 : Organização dos dados na TABCON.

Índice	Fil_1	Fil_2	Fil_3	Pai	D1	D2	D3	D4	Ct_1	Ct_2
1	2	3	0	1	5	4	65	20	1	4
2	0	0	0	1	7	6	63	18	5	9
3	4	5	6	1	1	1	80	24	10	16
4	0	0	0	3	8	8	72	17	17	19
5	0	0	0	3	20	10	50	15	20	24
6	0	0	0	3	1	20	80	24	25	31
.
.
.

Figura 6.15 : Organização dos dados na TABFORM.

índice	Sbnivel	Sbnivel 1	Sbnivel 2	Sbnivel 3	arquivo DAT	arquivo FRM	arquivo HLP
1	1	0	0	0	AQ1.DAT	AQ1.FRM	AQ1.HLP
2	1	2	4	6	AQ2.DAT	AQ2.FRM	AQ2.HLP
3	4	6	8	12	AQ3.DAT	AQ3.FRM	AQ3.HLP
4	6	9	15	18			
.
.
.

Figura 6.16 : Organização dos dados na TABACCESS.

Nro_descr	Descrição
1	CLASSE.
2	RELACIONAMENTO COM MEMBROS.
3	RELACIONAMENTO CLASSE A CLASSE.
4	TIPOS DE CLASSE.
5	PARAMETRO DE TEMPO.
.	.
.	.
.	.

Figura 6.17 : Organização dos dados na TABDESCR.

status	. . .						
1	2	3	4	5	6	7	. . .
2	5	7	8	10	12	13	. . .
2	20	23	25	26	69	72	. . .
2	30	33	35	37	38	39	. . .
2	30	47	49	50	52	55	. . .
.	
.	
.	

Figura 6.18 : Organização dos dados na TABFS.

A "tabmeta" no início da modelagem de uma aplicação encontra-se vazia, e cada elemento que tiver sido descrito, dependendo do sucesso obtido na modelagem, será gravado sua referência dos nós aceitos (índices) depois de ter sido feita uma consistência na "Tabfs" (Fig. 6.19).

solução	solução	solução	solução	solução	solução	. . .
1	2	3	4	5	6	. . .
2	5	7	8	10	12	. . .
2	20	23	25	26	69	. . .
2	30	33	35	37	38	. . .
2	30	47	49	50	52	. . .
.
.
.

Figura 6.19 : Organização dos dados na TABMETA.

Algoritmo PRINCIPAL DO DESCONT-THM.

=====

```

início { módulo principal }
  Carrega_estrutura;
  Busca_e_ativa_tarefas;
  Se tarefa_escolhida for manutenção_do_BC_THM
    então
      Consiste_senha
      Se senha_autorizada
        então
          Manutenção_do_BC_THM
        senão
          Modelagem_do_EC_THM
      fim se
    fim se
fim algoritmo. { fim módulo principal }

```

Algoritmo CARREGA ESTRUTURA.

=====

```
início
  Se status_carga diferente de carregada
    então
      Carregando_formatos
      Carregando_conteúdo
      Carregando_frames
      Carregando_niveis
      status_carga é igual a carregada
    fim se
fim algoritmo.
```

Algoritmo BUSCA_E_ATIVA_TAREFAS.

=====

```
início
  Mostra_opções_tarefa
  Enquanto tarefa_escolhida não for válida
    tarefa_escolhida é igual a opção
  fim enquanto
fim algoritmo.
```

Algoritmo MANUTENÇÃO_DO_BC-THM.

=====

```
início
  Enquanto opção_manutenção não for válida faça
    Leia opção_manutenção
    Case opção_manutenção de
      1 : Geração_de_arquivos_de_dados.
      2 : Geração_de_senha_de_usuários.
      3 : Ativa_edframe.
    fim case
  fim enquanto
fim algoritmo.
```

Algoritmo ATIVA_EDFRAME.

=====

```
início
  Ler_frame
  Trata_frame
  Se tecla_comando é gravar
    então
      Grava_frame
  fim se
fim algoritmo.
```

Em cada uma das etapas da função "MODELAGEM DO EC-THM" um procedimento será utilizado para tratar o grau de conhecimento que um usuário possui sobre o sistema.

Neste procedimento é estabelecida uma política de níveis, onde o usuário receberá, mais ou menos, detalhes acerca de conceitos ou exemplos do modelo THM.

Algoritmo POLÍTICA_DE_NÍVEIS.

=====

```
início
  Enquanto uma explanação for solicitada faça
    Se nível do usuário for 0
      então
        Exibe_explanação_nível_zero(indice,nível_usr)
        Se grau de satisfação da explanação
          for muito satisfeito
            então
              nível do usuário é 1
        fim se
    fim se
  Se nível do usuário for 1
    então
      Exibe_explanação_nível_um(indice,nível_usr)
      Se grau de satisfação da explanação
        for pouco satisfeito
          então
            nível do usuário é 0
      fim se
      Se grau de satisfação da explanação
        for muito satisfeito
```

```

        então
            nível do usuário é 2
        fim se
    fim se
Se nível do usuário for 2
    então
        Exibe_explanação_nível_dois(indice,nível_usr)
        Se grau de satisfação da explanação
        não for muito satisfeito
            então
                nível do usuário é 1
            fim se
        fim se
    fim se
fim enquanto
fim algoritmo.

```

6.2.1 Modelando classes.

Esta operação identificará os objetos que fazem parte do ambiente que se está modelando. O valor atribuído ao campo "status_m" representa o sucesso do estanciamento em determinado nó da taxonomia THM (ver Fig. 4.1) e armazenando em uma tabela da "Tabdat", cuja identificação corresponde também ao nó, assinalado como elemento da "Tabaccess".

Por exemplo: modelando classe número : 1.

```

identificação : "FONE".
status_m : 02 (número do nó na taxonomia).
tabela na "Tabdat" : CLASSE.DAT.
tabela na "Tabfrm" : CLASSE.FRM.
tabela na "Tabhlp" : CLASSE.HLP.

```

A figura 6.20 apresenta a organização interna da "Tabdat" (a instanciação de classes).

Índice	Status_m	Val_1
1	02	FONE
2	02	PREFIXO
3	02	NÚMERO
4	02	LUGAR
5	02	PESSOA
6	02	NOME
7	02	MULHER
8	02	HOMEM
9	02	QUANTIDADE
10	02	FAMÍLIA

Figura 6.20 : Dados armazenados na tabela CLASSE.DAT.

Quando uma classe for instanciada (após a realização dos testes de consistência realizados nos "frames") a mesma será armazenada na posição vazia da tabela "CLASSE.DAT", (sendo esta a ordem da classe); o status_m é o número do nó que está associado ao conceito na taxonomia e à identificação da classe.

O status_m de todas as tabelas que formam o "Tabdat" na ocasião de um valor aceito será avaliado na "Tabfs", onde procurará o grupo a que esta solução pertence. Em caso da localização do grupo de soluções, será gravado o status_m do elemento modelado na "Tabmeta" na posição que corresponde à classe "Tabdat".

Na sequência será instanciado um tipo que represente os elementos da classe.

Por exemplo: modelando os tipos de domínio da classe
número : 2.

identificação : "PREFIXO".

status_m : 75 (número do nó na taxonomia).

tabela na "Tabdat" : TP_DOMI.DAT.

tabela na "Tabfrm" : TP_DOMI.FRM.

tabela na "Tabhlp" : TP_DOMI.HLP.

A figura 6.21 apresenta a organização interna da "Tabdat"
(a instanciação do tipo de domínio).

Índice	Status_m	Val_1
1	75	A
2	75	I
3	75	I
4	75	M
5	75	M
6	75	S
7	75	G
8	75	G
9	75	I
10	75	S

Figura 6.21 : Dados armazenados na tabela TP_DOMI.DAT.

Em seguida serão modelados os tipos da classe (ainda na
função de modelagem de classe) sobre a representação de domínio
total ou parcialmente conhecido e, gravado os valores nas
tabelas da "Tabdat" apresentados nas figuras 6.22, 6.23 e 6.24.

Indice	Status_m	Val_1
1	72	C
.	-	-
.	-	-
6	72	C
7	72	C
8	72	C
.	-	-
10	-	-

Figura 6.22 : Dados armazenados na tabela COMPOSTO.DAT.

Indice	Status_m	Val_1
1	-	-
.	-	-
3	77	S
.	-	-
5	77	S
.	-	-
.	-	-
9	77	S
10	77	S

Figura 6.23 : Dados armazenados na tabela ESTATICO.DAT.

Indice	Status_m	Val_1
.	-	-
2	78	D
.	-	-
4	78	D
.	-	-
.	-	-
.	-	-
10	-	-

Figura 6.24 : Dados armazenados na tabela DINAMICO.DAT.

Algoritmo MODELAGEM_DO_EC-THM.

=====

```

inicio
  Enquanto escolha_modelagem não for válida faça
    Leia escolha_modelagem
    Case escolha_modelagem de
      1 : Modelagem_de_classe.
      2 : Modelagem_de_relacionamento.
      3 : Modelagem_de_tempo.
      4 : Relatório_e_geração_TECD.
      5 : Diagnóstico_finaliza.
    fim case
  fim enquanto
fim algoritmo.

```

Algoritmo MODELAGEM_DE_CLASSE.

=====

```
início
  Repita para sempre
    Localiza_posição_para_classe
    Recebe_nome_da_classe
    Processa_frame(indice, nome_da_classe)
    Se classe é aceita
      então
        Avalia_indice_na_Tabfs(indice)
        Se índice existe na Tabfs
          então
            Grava_dat(indice, nome_da_classe)
            Grava_meta(indice)
          fim se
        fim se
      fim repita
    fim algoritmo.
```

Algoritmo PROCESSA_FRAME(indice, nome_da_classe).

=====

```
início
  Enquanto nó não for raiz faça
    Se nó é terminal
      então
        Execute_PSLOTS(nome_da_classe)
        Enquanto nó tiver pai faça
          Execute_MSLOTS(nome_da_classe)
        fim enquanto
      fim se
    fim enquanto
  classe é verdadeiro
  fim algoritmo.
```

Algoritmo EXECUTE_PSLOTS(nome_da_classe).

=====

```
início
  Enquanto existir pslot faça
    Enquanto existir faceta faça
      Execute_faceta(nome_da_classe)
    fim enquanto
  fim enquanto
  fim algoritmo
```

Algoritmo EXECUTE_MSLOTS(nome_da_classe).

=====

```
inicio
  Enquanto existir mslot faça
    Enquanto existir faceta faça
      Execute_faceta(nome_da_classe)
    fim enquanto
  fim enquanto
fim algoritmo
```

6.2.2 Modelando relacionamentos.

A modelagem de relacionamentos compreende os relacionamentos hierárquicos (é-um, é-parte e é-elemento'), temporais (pré-pós) e relacionamentos comuns entre classes.

Algoritmo MODELAGEM_DE_RELACIONAMENTOS.

=====

```
inicio
  Enquanto escolha_relacionamento não for válida faça
    Leia escolha_relacionamento
    Case escolha_relacionamento de
      1 : Modela_generalização.
      2 : Modela_agregação.
      3 : Modela_agrupamento.
      4 : Modela_temporais.
      5 : Modela_rel_entre_classes.
    fim case
  fim enquanto
fim algoritmo.
```

6.2.2.1 Estabelecendo generalizações.

O usuário precisa identificar quais as classes que são generalizadas, atribuindo-lhes o nome do relacionamento à classe, armazenando valores na "Tabdat" conforme os nós que pertençam a esta função.

Como exemplo podemos observar que a classe "PESSOA" é uma classe generalizada, pois existe um papel aplicado à mesma, determinando que as classes "HOMEM e MULHER" serão identificados pelo relacionamento "é-um", ou seja, "HOMEM é uma PESSOA" e "MULHER é uma PESSOA".

As figuras 6.25, 6.26, 6.27, 6.28 e 6.29 detalham os valores que foram captados na identificação das generalizações provenientes do exemplo.

Índice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
5	33	e-um
.	-	-
.	-	-
.	-	-
10	-	-

Figura 6.25 : Dados armazenados na tabela GENER.DAT.

Índice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
5	35	sexo
.	-	-
.	-	-
10	-	-

Figura 6.26 : Dados armazenados na tabela NOMPAPEL.DAT.

Índice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
7	38	5
8	38	5
.	-	-
.	-	-
10	-	-

Figura 6.27 : Dados armazenados na tabela L_SUBCLA.DAT.

Índice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
5	41	E
.	-	-
.	-	-
10	-	-

Figura 6.28 : Dados armazenados na tabela TP_PAPEL.DAT.

Índice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
5	39	B
.	-	-
.	-	-
.	-	-
10	-	-

Figura 6.29 : Dados armazenados na tabela PARAMET.DAT.

Algoritmo MODELA_GENERALIZAÇÃO.

=====

```

início
  Repita para sempre
    Estabelece_ou_seleciona_classe
    Se classe existe
      então
        Avalia_indice_na_Tabfs(indice)
        Se indice existe na Tabfs
          então
            Classe_generalizada recebe classe
            Estabelece_papel à classe generalizada
            Identifica_lista_de_subclasse
            Define_tipo_ao_papel
            Define_relação_com_teorias_conjuntos
            Grava_dat(indice)
            Grava_meta(indice)
          fim se
        fim se
      fim repita
    fim algoritmo.

```

6.2.2.2 Estabelecendo agregações.

A modelagem de agregações identificará qual classe das que estão armazenadas na tabela "classe.dat", pertence ao

relacionamento hierárquico do tipo "é-parte-de", identificados como classes componentes de uma classe mais superior.

Como exemplo, é fácil perceber que a classe "FONE" é composta de "PREFIXO e NÚMERO", onde a classe "PREFIXO" é-parte de "FONE" e o mesmo acontecendo com "NÚMERO" (classes componentes).

As figuras 6.30, 6.31 e 6.32 demonstram a organização interna dos valores armazenados nas tabelas envolvidas durante a identificação da agregação.

Índice	Status_m	Val_1
1	47	e-parte
.	-	-
.	-	-
.	-	-
10	-	-

Figura 6.30 : Dados armazenados na tabela AGREG.DAT.

Índice	Status_m	Val_1
1	-	-
2	49	1
3	49	1
.	-	-
.	-	-
.	-	-
10	-	-

Figura 6.31 : Dados armazenados na tabela LCLACOMP.DAT.

Índice	Status_m	Val_1
1	55	1
.	-	-
.	-	-
.	-	-
10	-	-

Figura 6.32 : Dados armazenados na tabela AGRE_EXE.DAT.

Algoritmo MODELA_AGREGAÇÃO.

=====

```

inicio
  Repita para sempre
    Estabelece_ou_seleciona_classe
    Se classe existe
      então
        Avalia_indice_na_Tabfs(indice)
        Se indice existe na Tabfs
          então
            Classe_agregada recebe classe
            Estabelece_lista_de_classes_componentes
            Define_tipo_para_a_agregação
            Grava_dat(indice)
            Grava_meta(indice)
          fim se
        fim se
      fim se
    fim repita
  fim algoritmo.

```

6.2.2.3 Estabelecendo agrupamentos.

A modelagem de agrupamentos identificará qual das classes modeladas anteriormente permite estabelecer um relacionamento com uma classe elemento, formando, assim, uma classe que engloba a classe elemento.

As figuras 6.33, 6.34, e 6.35, apresentam as características armazenadas nas tabelas que compõem a modelagem de agrupamentos.

Indice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
.	-	-
10	58	e-elemento

Figura 6.33 : Dados armazenados na tabela AGRUP.DAT.

Indice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
5	60	10
.	-	-
.	-	-
.	-	-
10	-	-

Figura 6.34 : Dados armazenados na tabela CLA_ELEM.DAT.

Indice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
.	-	-
10	65	10

Figura 6.35 : Dados armazenados na tabela AGRU_EXP.DAT.

Algoritmo MODELA_AGRUPAMENTO.

=====

```
inicio
  Repita para sempre
    Estabelece_ou_seleciona_classe
    Se classe existe
      então
        Avalia_indice_na_Tabfs(indice)
        Se indice existe na Tabfs
          então
            Classe_agrupada recebe classe
            Estabelece_classe_elemento
            Define_tipo_para_agrupamento
            Grava_dat(indice)
            Grava_meta(indice)
          fim se
        fim se
      fim se
    fim repita
  fim algoritmo.
```

6.2.2.4 Estabelecendo relacionamentos comuns.

Os relacionamentos são modelados seguindo as restrições impostas pelo modelo THM com relação aos tipos hierárquicos e temporais. Será estabelecido um relacionamento entre classes (origem e destino), abrangendo as solicitações com respeito às cardinalidades ou tipos que envolvem os relacionamentos com relação a classe e membros, a composição de chaves e a manutenção de valores anteriores (entidades preservadas).

Por exemplo, o relacionamento "número-de" entre as classes "FONE e PESSOA" tem cardinalidade (1,1) e existe também um relacionamento entre "PESSOA e FONE" identificado como relacionamento "tem-fone" e cardinalidade (0,*) (relacionamento inverso especificado).

Caso não se especifique o nome do relacionamento entre uma classe origem (anteriormente destino) da mesma classe destino (anteriormente origem), será automaticamente especificado o nome do relacionamento como sendo o prefixo "INV" concatenado com a especificação anterior dada a classe destino e origem (especificação de um relacionamento inverso).

As figuras 6.36, 6.37, 6.38, 6.39, 6.40, 6.41 e 6.42 apresentam os dados que foram captados durante a modelagem dos relacionamentos comuns.

Indice	Status_m	Val_1	Val_2	Val_3
1	07	1	5	numero_de
2	07	5	1	tem_fone
3	07	1	4	em
4	07	4	1	inv_em
5	07	4	2	tem_prefixo
6	07	2	4	e_prefixo
7	07	5	6	tem_nome
8	07	6	5	inv_tem_nome
9	07	1	9	quantos
10	07	9	1	inv_quantos
11	07	7	8	tem_irmao
12	07	8	7	tem_irma
13	07	7	8	casada_com
14	07	8	7	casado_com

Figura 6.36 : Dados armazenados na tabela RELAC.DAT.

Indice	Status_m	Val_1
1	12	1
2	12	0
3	12	1
4	12	0
5	12	1
6	12	1
7	12	1
8	12	0
9	12	1
10	12	1
11	12	0
12	12	0
13	12	0
14	12	0

Figura 6.37 : Dados armazenados na tabela CARDMIN.DAT.

Indice	Status_m	Val_1
1	13	1
2	13	*
3	13	1
4	13	*
5	13	1
6	13	*
7	13	1
8	13	*
9	13	1
10	13	1
11	13	*
12	13	*
13	13	1
14	13	1

Figura 6.38 : Dados armazenados na tabela CARDMAX.DAT.

Índice	Status_m	Val_1	Val_2
1	16	S	2
.	-	-	-
.	-	-	-
.	-	-	-
14	-	-	-

Figura 6.39 : Dados armazenados na tabela PRE_REL.DAT.

Índice	Status_m	Val_1
1	18	M
2	18	M
3	18	M
4	18	M
5	18	M
6	18	M
7	18	M
8	-	-
9	18	M
10	18	M
11	18	M
12	18	M
13	18	M
14	18	M

Figura 6.40 : Dados armazenados na tabela TIP_CM.DAT.

Indice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
9	20	c
.	-	-
.	-	-
14	-	-

Figura 6.41 : Dados armazenados na tabela RCLACLA.DAT.

Indice	Status_m	Val_1
1	-	-
.	-	-
.	-	-
7	73	1
.	-	-
.	-	-
14	-	-

Figura 6.42 : Dados armazenados na tabela CHAVE.DAT.

Algoritmo MODELA_REL_ENTRE_CLASSE.

=====

```
inicio
  Repita para sempre
    Estabelece_ou_seleciona_classe
    Se classe existe
      então
        Avalia_indice_na_Tabfs(indice)
        Se indice existe na Tabfs
          então
            Estabelece_relacionamento
            Define_cardinalidade
            Identifica_chave
            Opção_manter_valores_anteriores
            Se opção confirmada
              então
                Valorar_quantidade
            fim se
            Distingue_tipo_relacionamento
            Grava_dat(indice)
            Grava_meta(indice)
          fim se
        fim se
      fim se
    fim repita
  fim algoritmo.
```

Algoritmo MODELAGEM_DE_TEMPO.

=====

```
inicio
  Repita para sempre
    Estabelece_ou_seleciona_classe
    Se classe existe
      então
        Avalia_indice_na_Tabfs(indice)
        Se indice existe na Tabfs
          então
            Verifica_se_existe_parametro_de_tempo
            Se parametro existe
              então
                Altera_parametro_tempo
              senão
                Atribui_parametro_de_tempo
            fim se
            Grava_dat(indice)
            Grava_meta(indice)
          fim se
        fim se
      fim se
    fim repita
  fim algoritmo.
```

Algoritmo ESTABELECE_OU_SELECIONA_CLASSE.
=====

```
início
  Ler classe definida pelo usuário
  Se classe do usuário não existir no classe.dat
  então
    Enquanto classe não for escolhida faça
      Mostra_classe
      Confirma classe escolhida
    fim enquanto
  fim se
fim algoritmo.
```

Algoritmo MODELA_TEMPORAIS.
=====

```
início
  Estabelece_ou_seleciona_classe
  Se classe existe
  então
    preclasse é igual à classe
    Estabelece_ou_seleciona_classe
    Se classe existe e classe <> preclasse
    então
      posclasse é igual a classe
      Define_rel_pré_pós
    fim se
  fim se
fim algoritmo.
```

Algoritmo RELATÓRIO_E_GERAÇÃO_TECD.
=====

```
início
  Enquanto não for fim da Tabmeta faça
    Leia registro da Tabmeta
    Enquanto solução_i <> 0
      Imprime arquivo_dat(solução_i)
      Gera_tecd(solução_i)
    fim enquanto
  fim enquanto
fim algoritmo.
```

Algoritmo DIAGNÓSTICO_FINALIZA.

=====

início

```
    Enquanto não for fim da Tabmeta faça
      Leia registro da Tabmeta
      Enquanto solução_i <> 0
        Localiza_solução na Tabfs
      fim enquanto
      Se solução_i da Tabmeta é igual a status_i a
        status_n da Tabfs
        então
          Imprime_diagnóstico_classe_completa
        senão
          Imprime_diagnóstico_falta_modelar
      fim se
    fim enquanto
fim algoritmo.
```

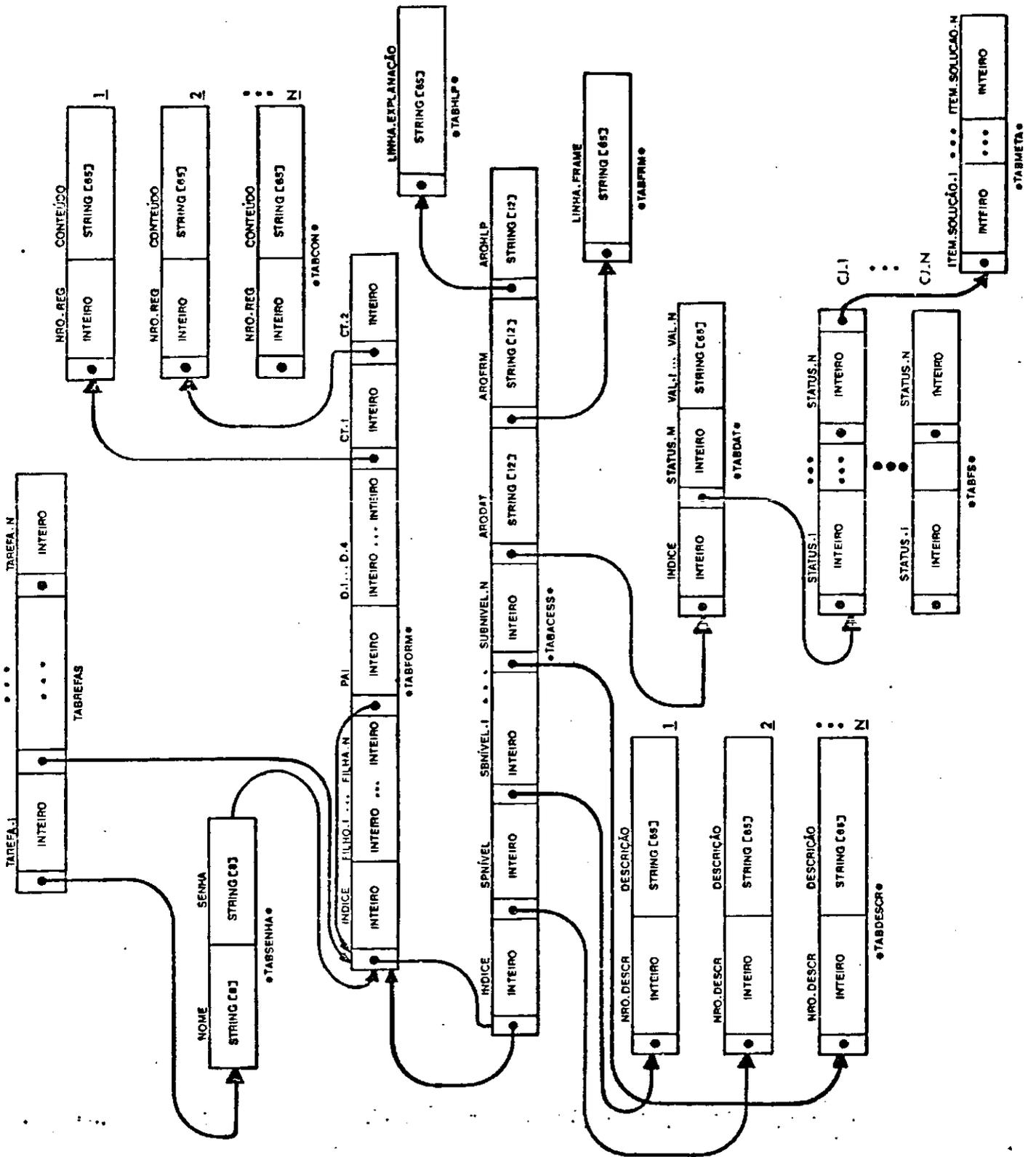


Figura 6.43 : Estrutura de dados que representam as construções arquiteturas do exemplo apresentado.

7. CONCLUSÕES E FUTUROS TRABALHOS

O principal objetivo deste trabalho é o de proporcionar aos usuários de diferentes níveis de conhecimento a formalização de sistemas de informação, e a geração do formalismo estático consistente de uma aplicação de esquema conceitual de dados.

A semântica da aplicação é retirada de uma porção do mundo real, através de processos automáticos que seguem uma metodologia, a qual é aplicada ao modelo semântico de dados THM, e representado em um BC que foi introduzido ao sistema.

O esquema conceitual gerado será posteriormente mapeado para o esquema interno relacional, sendo este o início da fase de implementação, desta feita sem o uso da LDD/THM.

O programa DESCNT-THM escrito em TURBO PASCAL, tem aproximadamente 9.000 linhas de código fonte, compilado na versão 3.1, gerando um código fonte no limite máximo da capacidade oferecida por esta versão. Foi deixada para o futuro a adaptação do programa DESCNT-THM para a versão 4.0 .

Para a evolução consequente do trabalho, deve-se pensar na integração desta ferramenta com as demais descritas no capítulo 1.

Seguindo a idéia da integração estabelecida para o sistema SISDECO-THM (em desenvolvimento), as TECDS serão utilizadas para consistências das tabelas de esquema conceitual de operações,

possibilitando a partir do SISDECO-THM executar funções do DESCNT-THM.

Notadamente, o TURBO PASCAL apresentou incongruências durante alguns procedimentos relacionados aos "frames" e, seria interessante utilizar para um BC baseado em frames, uma linguagem de programação orientada a objetos.

Outro trabalho que pode ser desenvolvido é uma ferramenta gráfica amigável que represente os aspectos estáticos de uma aplicação (utilizando simbologia conforme notação gráfica mostrado no apêndice C). Esta ferramenta geraria as TECDS a partir da estrutura gráfica adquirida da aplicação, ou vice-versa.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ANSI75] ANSI/X3/SPARC (Standards Planning and Requirements Committee), " Interino Report from the Study Group on Database Management System " FDT (Bulletin of ACM SIGMOD), Vol. 7, N. 2, 1975.
- [ATRE80] ATRE, S., " Data Base: Structured Techniques for Design, Performance, and Management". Addison-Wesley, New York, 1980.
- [BACH77] BACHMAN, C.W. " The Role Concept in Data Model ". Proc. 3rd International Conference on Very Large Database, Tokio, Japan, October 1977.
- [BPP76] BRACCHI, G.; P.PAOLINI; and G. PELAGATTI, " Binary Logical Association in Data Modelling " in J.M. Nijssen (ed), Modelling in Data Base Managment System (Proc. IFIP TC2 Conference, Freudenstadt), North-Holland, Amsterdam, The Netherlands, 1976.
- [BROD84] BRODIE, M.L. " On the Development of Data Models ". Topics in Information Systems: On Conceptual Modelling, Spring-Verlag, New York, 1984.
- [BW77] BOBROW, D & WINOGRAD, T. " Cognitive Science ". 1(1).1977. pag. 3 - 46. (in Reading's in KR-1986).
- [CT88] CARNOTA, R.J. & TESZKIEWICZ, A.D." Sistemas Expertos y Representacion del Conocimiento ". III EBAI Escola Brasileiro Argentina de Informática, Curitiba, Janeiro 1988.
- [CERI85] CERI, S. " Methodology and Tools for Data Base Design". North-Holland, Amsterdam 1985.

- [CHEN76] CHEN, P.P.S. " The Entity-Relationship Model: Toward a Unified View of Data ". ACM Transactions on Database Systems, Vol. 1, N. 1, March 1976.
- [CHIA88] CHIANCA, M.E.P.T. " Oftalmo: Um Sistema Especialista para Diagnóstico de Síndromes Oculares ". Dissertação de Mestrado, Departamento de Sistemas e Computação, UFPB, 1986.
- [CHIL77] CHILD, D. L. "Extended Set Theory". Proc. 3rd International Conference on Very Large Database, Tokio, Japan, October 1977.
- [CHU83] CHU, S.Y. " Banco de Dados: Organização, Sistemas, Administração". Atlas, São Paulo, 1983.
- [CODD70] CODD, E.F., " A Relational Model of Data for Large Shared Data Banks ". Communications of the ACM, Vol. 13, N. 6, June 1970, pp. 377 - 387.
- [CS85] CUNHA, J.F., SCHIEL, U. " Representação do Modelo Semântico de Dados THM ". Departamento de Sistemas e Computação da UFPB, Relatório Técnico, Campina Grande, Fevereiro 1985.
- [CUNH88] CUNHA, J.F. " Mapeamento do Modelo Semântico de Dados THM para o Modelo Relacional ". Dissertação de Mestrado, Departamento de Sistemas e Computação, UFPB, 1988.
- [DATE81] DATE, C.J., " An Introduction to Database Systems, 3rd ed., Addison-Wesley, Reading, Mass, 1981.
- [FERR87] FERREIRA, A. " Sistema de Efeitos Colaterais do Modelo THM ". Dissertação de Mestrado, Departamento de Sistemas e Computação, UFPB, 1987.

- [FK85] FIKES, R. & KEHLER, T. " The Role of Frame-Based Representation in Reasoning". Communications of the ACM. Division, 28(9): 904 - 920, Sept. 1985.
- [FS86] FURTADO, A.L & dos SANTOS, C.S., " Organização de Banco de Dados". Campus, Rio de Janeiro, 1986.
- [GRI82] GRIETHUYSEN, J.J. (ed), " Concepts and Terminology for the Conceptual Schema and Informations Base". ISO TC 97/SC5/WG3, 1982.
- [HSY85] HORNDASCH, A.A., STUDER, R., & YASDI, R., " An Approach to Conceptual Schema Design of Information System ". In: Working Conference on Theoretical and Formal Aspects on Information Systems, Sigtes, Espanha, North-Holland, 1985.
- [HUBB81] HUBBARD, G.U. " Computer-Assisted Data Base Design ". Van Nostrand Reinhold, New York, 1981.
- [JACO82] JACOBS, B.E. " On Database Logic ". Journal of the ACM, Vol. 29, N. 2, April 1982, pp 310 - 332.
- [KVIT88] KVITA, A.M. " Resolucion de Problemas con Inteligencia Artificial ". III EBAI Escola Brasileiro Argentina de Informática, Curitiba, Janeiro 1988.
- [MINS75] MINSKY, M. " A Framework for Representing Knowledge ". in P. Winston (eds.), The Psychology of Computer Vision, McGraw-Hill, New York, 1975, pp. 211 - 277.
- [MS85] MEDEIROS, C.L.G. & SCHIEL, U. " Projeto do Esquema Conceitual para Sistemas de Informações: Uma abordagem usando a metodologia THM ". Anais do SEMISH, Porto Alegre, 1986.

- [PASS87] PASSOS, E.P.L. " Sistemas com Base em Conhecimentos ".
VII Congresso da Sociedade Brasileira da
Computação - VI JAI - Salvador, Julho 1987.
- [SCHI82] SCHIEL, U. " The Temporal-Hierarchic Data Model ".
Bericht 10/82, Univ. Stuttgart, 1982.
- [SCHI84] SCHIEL, U. " Um Modelo Semântico de Dados e seu
Mapeamento para um Esquema Relacional Interno".
Tese de Doutorado, Univ. de Stuttgart (em alemão).
- [SETZ86] SETZER, V.W. " Banco de Dados: Conceitos, Modelo,
Gerenciadores, Projeto Lógico, Projeto Físico".
Edgard Blucher, São Paulo, 1986.
- [TURA87] TURAZI, A. " Sistema de Eventos e Triggers do Modelo
THM ". Dissertação de Mestrado, Departamento de
Sistemas e Computação, UFPB, 1987.
- [ULLM80] ULLMAN, J.D. " Principles of Database Systems ".
Computer Science Press, Potomac, Maryland, 1980.
- [WE79] WIEDERHOLD, G. & R. EL-MASRI, " The Structural Model
for Data Base Design ". Proc. International
Conference on the Entity-Relationship Approach to
Systems Analysis and Design, Los Angeles, Calif.,
December 1979.

APÊNDICE A

SINTAXE DA LINGUAGEM DE DEFINIÇÃO DE DADOS LDD/THM

A LDD/THM permite ao ABD definir a parte estrutural completa do ECD, incluindo a definição de classes com suas relações, papéis, agregações e agrupamentos. Neste apêndice está convencionado que as palavras reservadas são escritas em **negrito** e minúsculas (ver notação sintaxe na Fig. A1).

```
class <nome da classe>
  {class relationships}
    [<nome rel.> " ," <cl.relacionada>
      '(' <card_min> , <card_max> ')']*
    [pre-classe " ," <cl.relacionada>{exclusive}]*
    [post-classe " ," <cl.relacionada>{exclusive}]*

  {member relationships}
    [<rel-name> : <related class> '(' <card_min> <card_max> ') '
      {with old values}] +

  ((keys are ([<lista de chaves>]+ : inherited) :
    (type <tipo> {format <especificação>}))

  [<def.papel>]*
  [<def.agrupamento>]
  [<def.agregação>]

  {parameters: with time {and lifetime <constante> :
    <unidade de tempo>}}

<define papel> ::=
  with role <nome do papel>
    gives subclasses (<lista de classe> {explicit})
      by predicate <predicado>:
        using <relacionamento> as index
        parameters: (disjunctive : covering )

<def. agregação> ::=
  aggregation of (<lista de classes> (all : explicit):
    <classe-1, classe-2> by
    <relacionamento>] {exclusive})
```

```

<def.agrupamento> ::=
    grouping of (<nome da classe> (all : explicit) :
        grouping of (<nome da classe> (all : explicit):
            by predicate <predicado> :
            using <relacionamento>

```

Notação da Sintaxe	

text em negrito	- símbolo terminal
; : ,	- símbolos terminais
'string'	- o string é um terminal
[nome]	- não terminal
{texto}	- texto ocorre 0 ou 1 vez
[texto]*	- texto ocorre 0 ou mais vezes
[texto]+	- texto ocorre 1 ou mais vezes
:	- ou

Figura A1 : Notação da sintaxe da LDD/THM.

APÊNDICE B

SINTAXE DA LINGUAGEM DE MANIPULAÇÃO DE DADOS LMD/THM

Para a manipulação dos dados, a LMD/THM oferece operações primitivas para inserir ou remover determinada entidade em uma classe, mover uma entidade para outra, estabelecer relacionamento de uma entidade para outra, remover esse relacionamento ou atualizar, por remoção, o antigo e estabelecê-lo para outra entidade.

Usando essas operações e o princípio estrutural da LDD/THM, uma linguagem hierárquica de operações complexas pode ser construída.

As operações são iniciadas explicitamente pelo usuário ou chamadas por operações explícitas. A notação da sintaxe é mostrada na figura A1 (a mesma adotada para a LDD/THM).

```
operation <nome operação>
{input parameters
  [<nome parâmetro> : <nome classe> (external):
    from DB { by Key <lista chaves>]+}
{output-parameters
  [<nome parâmetro> : <nome classe> (external:
    to DB {by Key <lista chaves>]+}
{pre_conditions
  [(<predicado>{at<tempo>} {otherwise (warning:cancel:error)
    {<mensagem>}}> : <cláusula let>)+}
```

body

<declaração conjutiva>:
<declaração disjuntiva>:
<declaração interativa>:

{pos_conditions
idem pre_conditions }

<declaração conjutiva> ::=
[<declaração> {only if <predicado>}]+

<declaração disjuntiva> ::=
case [<predicado> : <declaração>]+

<declaração interativa> ::=
for each <condição> do <declaração conjutiva>

<condição> ::= ((var-entidade>in(<entidade de grupo>:<classe>)):
class <var-classe>) {such that <predicado>}

<declaração> ::= <operação primitiva> : <operação call>:
<cláusula let>

<operação primitiva> ::= (<insert> : <gr-insert> :
<gr-delete> : <nome> : <establish> : <remove> :
<update>) { at<tempo>}

<operação call> ::= <nome operação> "("<lista parâmetros
entrada>; <lista parâmetro saída>

<cláusula let> ::= let<variável> be<termo>

<insert> ::= insert<termo> into<classe>

<delete> ::= delete {<termo> from <classe>}

<gr-insert> ::= gr-insert<termo> into <entidade-de-grupo>

<gr-delete> ::= gr-delete<termo> from <entidade-de-grupo>

<move> ::= move<termo> from <classe1> to <classe2>

<establish> ::= establish (<termo1> : <classe>) <relacionamento>
<termo2>

<remove> ::= remove(<termo1> : <classe>) <relacionamento>
<termo2>

<update> ::= update(<termo1> : <classe>) <relacionamento>
<termo2> to <termo3>

```

<tempo> ::= (clock.<unidade>{operador aritmético}
  <valor numérico>}) : <especificação unidade>:
  <tupla de tempo> : <intervalo de tempo>

<unidade> ::= "day" : "month" : "year" : "hour" : "minute" :
  "second"

<especificação da unidade> ::= <unidade>: <valor>

<intervalo de tempo> ::= "<" <tempo> to <tempo> ">"

<tupla de tempo> ::= <ano><mês><hora><minuto><segundo>

<predicado> ::= <predicado primitivo> : <termo1>
  <operador relacional> <termo2> : not <predicado> :
  if<predicador1> then <predicador2>
  <predicador1> or <predicador2> :
  <relacionamento de membros> "("<termo1>,<termo2>)"

<termo> ::= <constante> : <variável> : <função> :
  <termo1><operador aritmético> <termo2> : <termo>

<função> ::= <relacionamento membros> "(" <termos> ")" :
  <relacionamento classe> "(" " ")"

<predicado primitivo> ::= in : is-rel : is-a : role-comp ...

<in> ::= <termo> in <classe>

<is-part> ::= is-part "("<entidade simples>,
  <entidade agregada> ")"

<nome operação> ::= <Identificador>

<nome parâmetro> ::= <Identificador>

<nome classe> ::= <Identificador>

<lista chaves> ::= [<Identificador>]

<variável> ::= <Identificador>

<constante> ::= <numero> : <string>

<entidade> ::= <var-entidade> : constante

<var-entidade> ::= "X" : "Y" : "Z"

<entidade de grupo> ::= <Identificador> : <constante>

<lista param entrada> ::= <Identificador> : <constante>
  [", "<lista param>]*

```

<lista param saída> ::= Idem param entrada.
 <relacionamento> ::= <relacionamento de membros> :
 <relacionamento de classe>
 <op. aritmético> ::= + : - : * : /
 <valor numérico> ::= <dígito> [<dígito>]*
 <op. relacional> ::= < : > : <= : >=
 <relacionamentos de membros> ::= <termo1> <rel.memb> <termo2>
 <relacionamento de classe> ::= <classe> <rel.classe> <termo>
 <rel. memb> ::= <Identificador>
 <rel.classe> ::= <Identificador>
 <entidade agregada> ::= "<" <termo> ["," <termo>]*
 <classe> ::= <Identificador>
 <classe agregada> ::= <Identificador>
 <classe componente> ::= <Identificador>
 <classe elemento> ::= <Identificador>
 <classe generalizada> ::= <Identificador>
 <subclasse> ::= <Identificador>
 <identificador> ::= <letra> [letra : dígito]*
 <letra> ::= A,B...Z
 <dígito> ::= 0,1...9

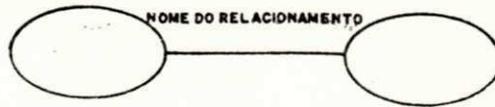
APÊNDICE C

FORMALISMO GRÁFICO DO MODELO THM

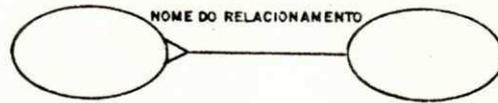
CLASSE



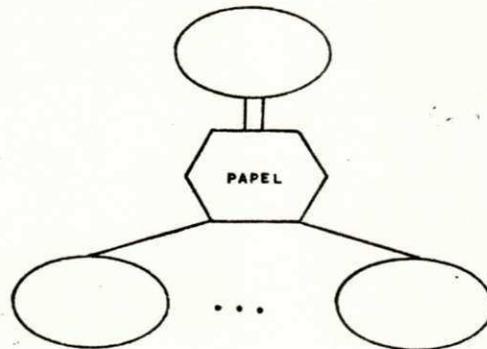
RELACIONAMENTO DE MEMBRO



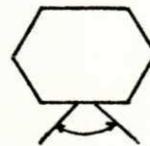
RELACIONAMENTO DE CLASSE



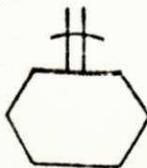
GENERALIZAÇÃO (PAPEL)



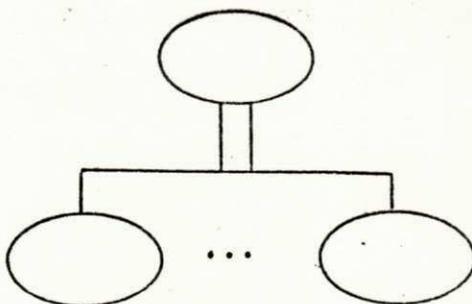
SUBCLASSES DISJUNTAS



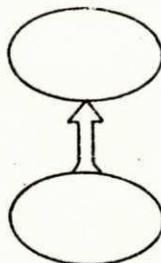
PAPEL COBRINDO



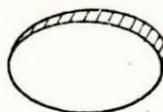
AGREGAÇÃO



AGRUPAMENTO



CLASSE COM TEMPO



RELACIONAMENTO PRÉ SIMPLES / PÓS SIMPLES



RELACIONAMENTO PRÉ EXCLUSIVO / PÓS SIMPLES



RELACIONAMENTO PRÉ SIMPLES / PÓS EXCLUSIVO



RELACIONAMENTO PRÉ EXCLUSIVO / PÓS EXCLUSIVO



APÊNDICE D

DESCRIÇÃO DAS TABELAS DE ESQUEMAS CONCEITUAIS DE DADOS

As tabelas de esquemas conceituais de dados armazenam os dados referentes à modelagem de uma aplicação do UD, e sua estrutura é resultado da execução do sistema DESCONT-THM, que tem em seu elenco de funções a de substituir a linguagem de definição de dados (LDD/THM).

O conjunto de tabelas fornecidas pelo sistema DESCONT-THM são: classes, relacionamentos, generalizações e papéis, agregações e agrupamentos.

D.1 Tabela de classe.

CLASSE : (NOME CLASSE, TIPO CLASSE, TIPO ENTIDADE, TEMPO, TEMPO DE VIDA, EVENTO, EXCLUIDA).

Nome da classe - indica o nome da classe.

Tipo classe - indica se a classe é de domínio (S=estática ou D=dinâmica) ou uma classe geral (C=composta).

Tipo entidade - indica o tipo da entidade. Além dos três tipos padrões (inteiro, real, string) podemos ter:

"A" - para entidade agregada.

"M" - para uma entidade identificada por relacionamento de membros.

"G" - para a identificação de uma entidade de subclasse gerada por uma classe generalizada.

Tempo - indica se é uma classe com intervalo de tempo (S/N).

Tempo de vida - indica o tempo de vida para entidades desta classe.

Evento - informa se para esta classe existem eventos da forma " on insert " ou " on delete ".

Excluída - Mostra se existe para a classe uma relação própria no banco de dados. A princípio toda classe gera uma E_relação que pode ser eliminada.

D.2 Tabela de relacionamentos(membros e classes).

RELACIONAMENTOS (NOME RELACIONAMENTO, CAR.MIN, CAR.MAX, CLASSE ORIGEM, RELAC INVERSO, CARMIN.INV, CARMAX.INV, CLASSE RELACIONADA, IDENTIF.CHAVE, VAL.ANTER, QUANT.VAL, TIPO RELAC, EVENTO, EXCLUÍDA).

Nome relacionamento - indica o nome do relacionamento.

Card.min, Card.max - indicam respectivamente a

cardinalidade mínima e a
cardinalidade máxima do
relacionamento.

Classe origem - indica a classe que dá origem ao
relacionamento.

Relac.inverso - indica o relacionamento inverso.
No caso de não existir
relacionamento inverso, este fica
representado colocando-se o prefixo
INV antes do nome do relacionamento.

Cardmin.inv
e
Cardmax.inv - indicam, respectivamente, a
cardinalidade mínima e máxima do
relacionamento inverso.

Classe relacionada - indica a classe para a qual existe o
relacionamento.

Identif.chave - indica se o relacionamento é ou não
chave da classe origem, podendo
assumir os seguintes valores:

0 - o relacionamento não é chave nem
parte da chave.

1 - o relacionamento é chave
principal da classe origem.

2 - o relacionamento é parte da chave principal da classe origem.

- Val. anter - indica se, para o relacionamento ficam registrados valores anteriores.
- Quant.val - indicam a quantidade de valores anteriores que devem ser conservados (valor inteiro).
- Tipo relac - distingue se o relacionamento é de classe ou de membro.
- Evento - indica se existem, para o relacionamento, eventos da forma " on establish " ou " on remove ".
- Excluida - indicam se o relacionamento inverso está armazenado no esquema. A princípio todos os relacionamentos 1:1 e n:1 passam a ser atributos, podendo ser eliminados.

D.3 Tabela de relacionamentos temporais (PRÉ-PÓS).

PRÉ-PÓS (PRÉ-CLASSE, PÓS-CLASSE, PRÉ-PÓS)

Pré-classe - dá o nome da classe da qual sai o relacionamento temporal.

Pós-classe - indica a classe para a qual existe o relacionamento temporal.

Pré-pós - Podendo assumir quatro valores:

"ss" - o relacionamento é
Pré-simples/Pós-simples.

"se" - o relacionamento é
Pré-simples/Pós-exclusivo.

"es" - o relacionamento é
Pré-exclusivo/Pós-simples.

"ee" - o relacionamento é
Pré-exclusivo/Pós-exclusivo.

D.4 Tabela de papéis.

PAPEL (NOME PAPEL, CLASSE, TIPO, REL CONJUNTO)

Nome papel - indica o nome do papel.

Classe - indica o nome da classe na qual o papel é aplicado.

Tipo - pode assumir os seguintes valores:

"E" - os elementos das subclasses resultantes são indicados explicitamente.

"P" - as subclasses são determinadas por um predicado.

"R" - as subclasses são determinadas por um relacionamento de membros.

Rel.conjunto - indica como a generalização está relacionada com a teoria dos conjuntos:

"D" - as subclasses devem ser disjuntas.

"C" - a união das subclasses devem formar a classe generalizada.

"B" - "D" e "C".

" " - não especificado.

D.5 Tabela de generalização.

GENERALIZAÇÕES (SUBCLASSE, CLASSE GENERALIZADA, PAPEL)

Subclasse - indica o nome da classe gerada pela aplicação do papel.

Classe generalizada - indica o nome da classe generalizada.

Papel - indica o papel aplicado á classe generalizada.

D.6 Tabela de agregação.

AGREGAÇÕES (CLASSE COMPONENTE, CLASSE AGREGADA, TIPO)

Classe componente - dá o nome de uma classe componente dentro de uma agregação.

Classe agregada - dá o nome da classe agregada.

Tipo - pode assumir os seguintes valores:

"A" - é formado todo o produto cartesiano sobre as classes componentes.

"E" - é deixado ao usuário a decisão de quais entidades das classes componentes ele reunirá em uma entidade agregada.

D.7 Tabela de agrupamentos.

AGRUPAMENTOS (CLASSE ELEMENTOS, CLASSE GRUPOS, TIPOS)

Classe elementos - dá o nome da classe de cujas entidades são formados os grupos.

Classe grupos - nome da classe grupos.

Tipo - assume os seguintes valores:

"A" - identifica a classe de grupo

como conjunto da classe de elementos.

"E" - indica que os grupos devem ser formados explicitamente pelo usuário.

"R" - os grupos são formados com base em um relacionamento de membros.

"P" - indica que o agrupamento é feito com base em um predicado.

APÊNDICE E

TUTORIAL DO SISTEMA DESCNT-THM

E.1. INSTALANDO O DESCNT-THM EM SEU COMPUTADOR.

Configuração básica do equipamento:

- (a) 1 unidade de disco flexível.
- (b) 640 Kbytes de memória.

E.1.1 - Instalação do sistema DESCNT em disco virtual.

E.1.1.1 - Preparando seu disco de carga do sistema operacional.

O DESCNT-THM necessita de um arquivo CONFIG.SYS no seu disco de carga do DOS (o disco que você usa para inicializar seu computador).

O CONFIG.SYS é composto dos seguintes comandos:

```
FILES          = 20
BUFFERS        = 20
VDISK 360 512 64
```

Obs: O disco do DOS deve conter o arquivo VDISK.SYS, que é responsável pela geração do disco virtual em seu equipamento.

E.1.1.2 - Copiando o DESCNT-THM para um disco virtual.

Insira o disco de PROGRAMAS (DISCO N. 1) em uma unidade de disco (unidade A) e copie os arquivos deste disco para a unidade de disco virtual (unidade C) que foi gerada no processo de carga do DOS:

- . Inserir o disco de PROGRAMAS na unidade A.
- . Digitar "Copy A:*. * C:" <ENTER>
- . Digitar C: <ENTER>
- . Inserir o disco de DADOS (DISCO N. 2) na unidade A.

E.1.2 - Instalação do sistema DESCNT-THM em disco rígido.

Insira o disco de PROGRAMAS (DISCO N. 1) na unidade de disco A e copie os arquivos deste disco para a unidade de disco rígido (unidade D).

- . Inserir o disco de PROGRAMAS na unidade A.
- . Digitar "COPY A:*. * D:" <ENTER>
- . Digitar D: <ENTER>
- . Inserir o disco de DADOS (DISCO N. 2) na unidade A.

E.1.3 - Ativando o sistema DESCNT-THM.

Para ativar o sistema, estabeleça "C:" como unidade corrente, onde se encontram os arquivos de "PROGRAMAS", e digite "DESCNT".

Ao ativar o sistema DESCONT-THM, será exibida uma tela informando a versão e local de desenvolvimento do sistema, mostrando em seguida uma tela que permitirá ao usuário [C]ARREGAR as estruturas do DESCONT-THM (Tabform, Tabcon, Tabaccess, Tabdescr) ou [S]AIR para o sistema operacional.

Ao optar por [C]ARREGAR, as estruturas do DESCONT-THM serão armazenadas na memória, diminuindo o tempo de acesso a estas estruturas em qualquer das funções do DESCONT-THM.

Após ser feita a carga das estruturas será apresentada uma tela de Menu Principal (Fig. E.1).

Neste ponto, o usuário poderá optar por fazer a MANUTENÇÃO DO BC-THM, MODELAGEM DO EC-THM ou SAÍDA DO SISTEMA DESCONT-THM para o sistema operacional.

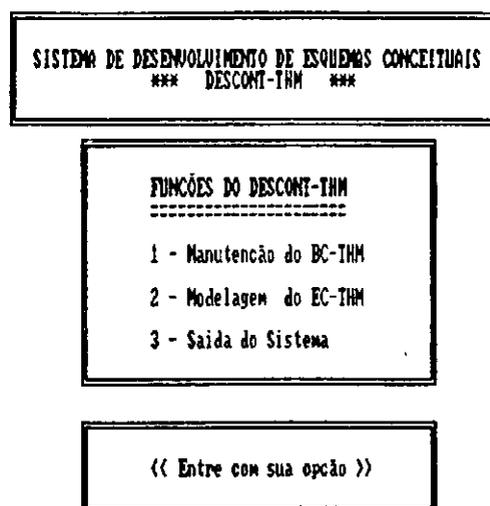


Figura E.1: Menu principal do DESCONT-THM.

E.2. MANUTENÇÃO DO BC-THM.

Este módulo tem a função de gerar arquivos e registros para controles e modificar um BC existente ou editar completamente um BC através do EDFRAME.

É apresentada uma tela (Fig. E.2) que permitirá ao usuário a geração dos "Arquivos de Dados" e das "Senhas dos Usuários" do DESCONT-THM, a "MANUTENÇÃO do BC-THM" ou "Fim da operação".

E.2.1 Geração de Arquivos de Dados de DESCONT-THM

Esta seção é responsável pela geração de um conjunto de arquivos onde serão armazenados os valores instanciados durante a seção de modelagem do EC-THM.

SISTEMA DE DESENVOLVIMENTO DE ESQUEMAS CONCEITUAIS *** DESCONT-THM ***
1 - Geracao dos Arquivos de Dados do DESCONT-THM 2 - Geracao das Senhas dos usuarios do DESCONT-THM 3 - Manutencao do BC-THM 4 - Fim de Operacao
<< ***** Escolha sua opcao ***** >>

Figura E.2: Funções de Manutenção do DESCONT-THM.

Serão mostradas as telas (Figs. E.3 e E.4) contendo o mapa geral de arquivos a serem gerados.

```
SISTEMA DE DESENVOLVIMENTO DE ESQUEMAS CONCEITUAIS
*** DESCONT-THM ***

*** MAPA GERAL DOS ARQUIVOS PARA GERACAO ***

1-CLASSE.DAT   2-RELAC.DAT   3-NOMEREL.DAT  4-CLASSERE.DAT
5-CARDINA.DAT  6-CARDMIN.DAT 7-CARDMAX.DAT  8-TIP_PM.DAT
9-PRE_REL.DAT 10-TIP_CM.DAT 11-RELACLA.DAT 12-PREPOS.DAT
13-PP_CLREL.DAT 14-EX_POS.DAT 15-EX_PRE.DAT  16-EXPREPOS.DAT
17-HIERARQ.DAT 18-GENER.DAT  19-PAPEL.DAT  20-NOMPAPEL.DAT
21-L_SUBCLA.DAT 22-PARAMET.DAT 23-TP_PAPEL   24-GEN_EXPL.DAT

(( Informe o numero do Arquivo para Geracao ))
(( Para continuacao do mapa tecla (ENTER) ))
```

Figura E.3: Mapa geral dos arquivos para geração.

```
SISTEMA DE DESENVOLVIMENTO DE ESQUEMAS CONCEITUAIS
*** DESCONT-THM ***

*** MAPA GERAL DOS USUARIOS PARA GERACAO ***

25-GEN_PRED.DAT 26-GEN_USR.DAT 27-AGREG.DAT  28-LCLACOMP.DAT
29-EXCLUSIV.DAT 30-TP_AGREG.DAT 31-AGRE_ALL.DAT 32-AGRE_EXP.DAT
33-AGRE_USR.DAT 34-AGRUP.DAT   35-CLA_ELEM.DAT 36-TP_AGRUP.DAT
37-AGRU_ALL.DAT 38-AGRU_EXP.DAT 39-AGRU_PRED.DAT 40-AGRU_USR.DAT
41-MEMB_CLA.DAT 42-COMPOSTA.DAT 43-CHAVE.DAT   44-TP_DOMIN.DAT
45-ESTATICO.DAT 46-DINAMICO.DAT 47-PAR_TEMP.DAT 48-SENHA.DAT

(( Informe o numero do Arquivo para Geracao ))
(( Para Cancelamento da Geracao tecla [C] ))  continuacao do mapa
```

Figura E.4: Continuação do mapa dos arquivos.

O usuário poderá escolher itens do mapa para realizar a geração física do arquivo, ou cancelar a operação.

Ao ser estabelecida uma escolha, e esta confirmada, será executada a geração, apresentando as mensagens resultantes da respectiva situação de geração (as mensagens do sistema são descritas no apêndice F).

E.2.2 Geração de Senhas dos usuários do DESCONT-THM.

Será acionada por um usuário habilitado (com registro na tabela de senhas).

O usuário terá cinco oportunidades para que sua habilitação seja concedida pelo sistema, caso contrário, o sistema será desativado.

A figura E.5 ilustra um exemplo de catalogação de senha de usuário na "Tabsenha".

```
SISTEMA DE DESENVOLVIMENTO DE ESQUEMAS CONCEITUAIS
*** DESCONT-THM ***
```

Gerando SENHA para BC-THM

Qual o Nome ? LAURINDO

Entre com a SENHA ?

Figura E.5: Gerando senhas para o BC-THM.

E.2.3 - Manutenção do BC-THM.

Realiza um processo navegacional de localização para o

acesso a um "frame" do BC-THM através da ativação do EDFRAME. Para a realização do processo navegacional da "MANUTENÇÃO DO BC-THM", o usuário é submetido ao processo de teste e validação de sua habilitação para acessar "frames".

Quando uma habilitação é confirmada, o processo é iniciado a partir do "Menu Principal" conforme a figura E.6.

A disposição das telas durante o processo navegacional obedece com rigor à hierarquia do modelo THM (ver no capítulo 4 Fig. 4.1), cujo objetivo é atingir um elemento "frame" (representando um elemento terminal).

BANCO DE CONHECIMENTO * T H M *
==> MENU PRINCIPAL <==

1. Classe 2. Sub_Frame de Classe 3. Saída do Sistema
Entre com sua escolha :

Figura E.6: Início do processo navegacional para acessar frames.

A tela mostrada na figura E.7 ilustra um elemento "frame" (CLASSE) cujas opções oferecidas ao usuário são:

[A]ACESSO AO FRAME

[R]ETORNAR (ao elemento superior (um pai) de um elemento frame).

BANCO DE CONHECIMENTO * T H M *
==> IDENTIFICAÇÃO DA CLASSE <== ----- Frame : Classe
ACESSO AO FRAME Tecla < A > PARA RETORNAR Tecla < R >

Figura E.7: Tela de acesso ao frame.

Se a opção de acesso for escolhida, será exibida uma tela mostrando os superníveis e subníveis do "frame" dentro da taxonomia do BC-THM, enquanto ativa o editor "EDFRAME" e carrega os "SLOTS" do "frame" que se está acessando.

A figura E.8 apresenta um exemplo da edição do "frame", classe que foi acessada pela opção acesso mostrada na figura E.7.

```

===> Status do Editor de Frames ===>      Alt      Tot: 18 Lin: 1 Col: 1

FRAME          : Classe.
SUPERNIVEL     : Classe.
SUBNIVEIS      : Relacionamento-com Membros.
                : Relacionamento-Classe-a-Classe.
                : Tipos-de-Classe.

M-SLOT         : M-Restricao-da-Classe.
FAC1           : Nao-Numerico.
FAC2           : Classe-Unica.
FAC3           : Tamanho-Max-25.
FAC4           : Desconhecido.
                { Se-Necessario
                SN1 : Transforma-Maiuscula.
                SN2 : Troca-Branco-Sublinhado.

P-SLOT         : P-Variavel-da-Classe.
FAC1           : Varclasse.

```

Figura E.8: Edição de um frame.

A figura E.9 ilustra "Sub-frames de Classe", que também poderão ser acessados quando escolhidos dentro do processo navegacional em uma estrutura do BC-THM.

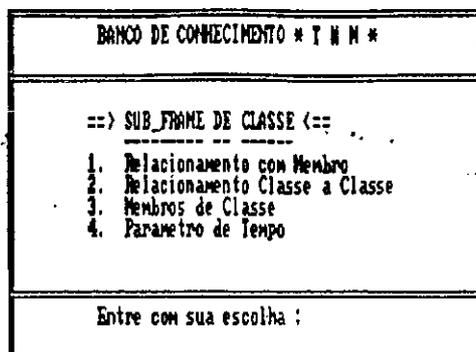


Figura E.9: Exemplo de subníveis não terminais do frame classe.

E.3. MODELAGEM DO EC- THM

Ao iniciar a "MODELAGEM DO EC-THM", é apresentada uma tela, solicitando ao usuário informações sobre o grau de conhecimento quanto ao uso do sistema (Fig. E.10).

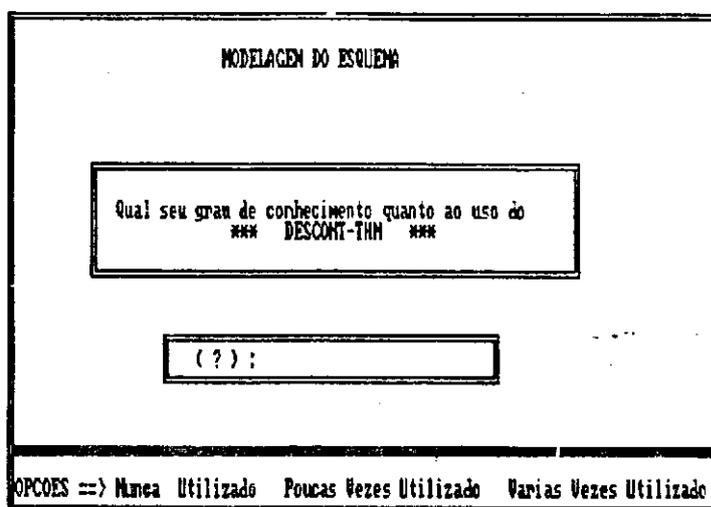


Figura E.10: Grau de conhecimento do usuário.

As opções especificam níveis de conhecimento, e consequentemente, influirão nos níveis de abordagem que o sistema faz durante o acesso e execução do módulo de explanação.

Em seguida é mostrada nova tela (Fig. E.11), onde são apresentadas todas as funções da modelagem do EC-THM, que são necessárias para especificar os aspectos estáticos de uma aplicação.

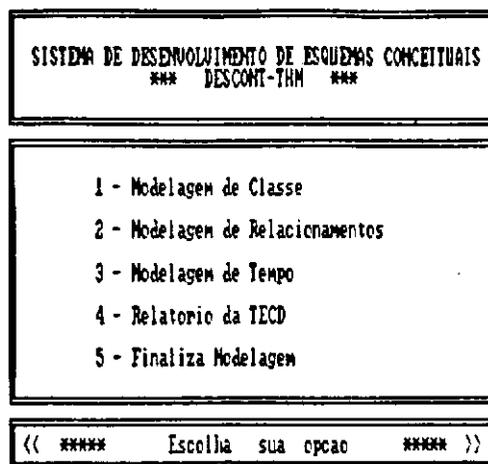


Figura E.11: Funções para modelagem do EC-THM.

E.3.1 - Modelagem de Classe.

Ao iniciar a modelagem de uma classe, é apresentada uma tela (Fig.E.12) contendo o número da classe que será modelada e o usuário especificará o nome da classe.

No caso da classe ser aceita pelo sistema, será mostrada uma tela (Fig. E.13) para obtenção da identificação do tipo (inteiro, real ou string) das entidades da classe.

MODELAGEM DE CLASSE

MODELANDO CLASSE Nº 21

QUAL O NOME DA CLASSE, OBJETO OU COISA ?

CARRO

Figura E.12: Modelando uma classe.

MODELAGEM DE CLASSE

Escolha um dos tipos de <OPCOES> que possa
representar as entidades (elementos) da classe

PECAS

(?) :

OPCOES ==> Inteiro Real String

Figura E.13: Identificando os tipos de uma classe.

Em seguida, o sistema apresenta uma tela onde questiona se as entidades da classe têm um domínio total ou parcialmente conhecido (estático ou dinâmico).

Obs.: Caso a classe não seja aceita pelo sistema (nome da classe

inconsistente), será apresentada uma mensagem de erro, facultando ao usuário ativar o módulo de explanação, pressionando a tecla "F1" (Fig. E.14).

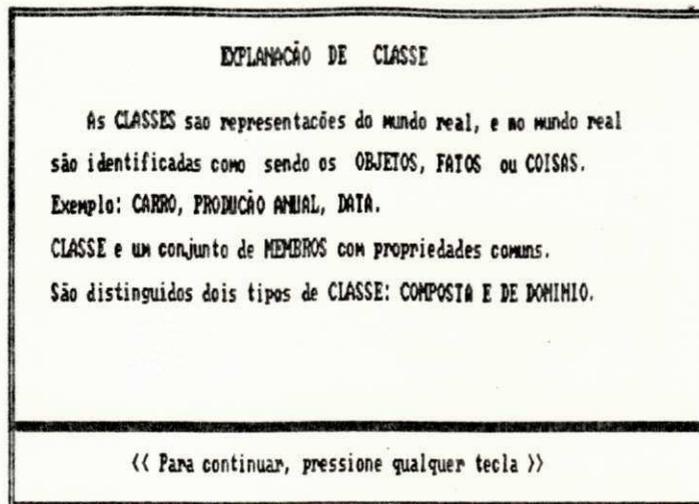


Figura E.14: Uma explanação solicitada.

Caso o módulo de explanação seja ativado, serão apresentados detalhes da referida modelagem que se está realizando, com o nível de conhecimento do usuário (determinado pelo sistema na política de níveis). Ao desativá-lo, o sistema apresenta uma tela (Fig. E.15) onde obtém o grau de satisfação do usuário quanto ao uso da explanação.

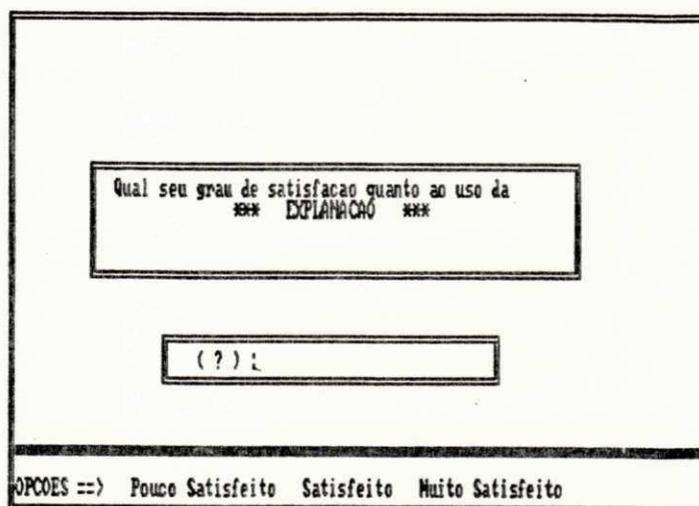


Figura E.15: Grau de satisfação para política de nível.

E.3.2 - Modelagem de Relacionamentos.

É apresentada uma tela, mostrada na figura E.16, na qual o usuário escolherá o tipo de relacionamento a ser modelado, ou o mesmo sairá da modelagem de relacionamento.

SISTEMA DE DESENVOLVIMENTO DE ESQUEMAS CONCEITUAIS *** DESCONT-TM ***	
1 - Classe1 (Relacionamento E_UM com)	Classe2
2 - Classe1 (Relacionamento E_PARTE da)	Classe2
3 - Classe1 (Relacionamento E_ELEMENTO da)	Classe2
4 - Classe1 (Uma descricao qualquer com)	Classe2
5 - Classe1 (Relacionamento TEMPORAL com)	Classe2
6 - Saida da Modelagem de Relacionamento	
<< **** Escolha sua opção **** >>	

Figura E.16: Funções da modelagem de relacionamentos.

E.3.2.1 - Relacionamento hierárquico do tipo E_UM.

É apresentada uma tela (Fig. E.17) na qual o usuário fornecerá o nome da classe generalizada.

MODELAGEM DE GENERALIZACAO
Qual o nome da classe generalizada?

Figura E.17: Modelando generalizações.

Caso o usuário não forneça nenhum nome (e teclar <ENTER>) o sistema apresentará uma lista de classes que foram modeladas anteriormente para que seja feita uma escolha.

No caso da classe ser aceita pelo sistema, será mostrada uma tela (Fig. E.18) na qual o usuário fornecerá o nome do papel aplicado à classe.

A seguir, o usuário deverá confirmar se os elementos das subclasses são determinados por um predicado ou por um relacionamento com membros.

Logo após, o sistema perguntará se as subclasses são disjuntas e se a união das subclasses geram a classe generalizada.

MODELAGEM DE GENERALIZACAO

Qual o nome do papel aplicado a classe ?

A imagem mostra uma tela de software com um título "MODELAGEM DE GENERALIZACAO" no topo. Abaixo do título, há um campo de entrada de texto contendo a pergunta "Qual o nome do papel aplicado a classe ?".

Figura E.18: Identificando um papel.

3.2.2 - Relacionamento hierárquico do tipo E_PARTE.

É apresentada uma tela (Fig. E.19) na qual o usuário

fornecerá o nome da classe agregada.

MODIFICAÇÃO DE AGREGAÇÃO

Qual o nome da classe agregada

Figura E.19: Modelando agregações.

No caso de não ter sido fornecido nenhum nome, o sistema apresentará a lista de classes que foram modeladas anteriormente.

Se a classe for aceita pelo sistema, será pedido o nome da classe componente. Caso o usuário não forneça nenhum nome, o sistema apresentará a lista de classes modeladas anteriormente. Ao aceitar uma classe, o sistema questionará se é formado todo o produto cartesiano sobre as classes componentes.

E.3.2.3 - Relacionamento hierárquico do tipo E_ELEMENTO.

É apresentada uma tela (Fig. E.20) na qual o usuário fornecerá o nome da classe que é elemento da classe agrupada.

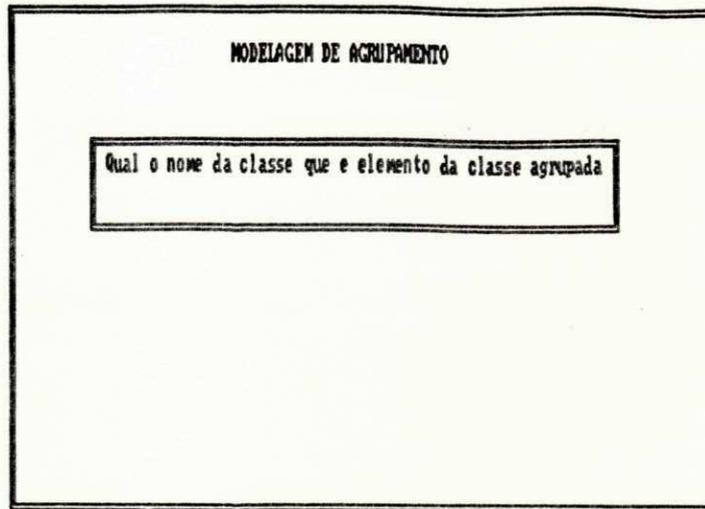


Figura E.20: Identificando classes elementos.

Sendo a classe aceita pelo sistema, será apresentada uma tela, solicitando o nome da classe agrupada, seguido de um pedido de confirmação.

Em seguida, o usuário deverá confirmar se a classe de grupo é um conjunto de classes de elementos ou se os grupos devem ser formados explicitamente pelo usuário ou são formados com base em um relacionamento de membros, ou são feitos com base em predicados.

E.3.2.4 - Uma descrição qualquer com uma classe destino.

É apresentada uma tela (Fig. E.21) na qual o sistema solicitará o nome da classe origem do relacionamento. Logo após, serão requeridos os nomes da classe destino e do relacionamento.

MODELAGEM DE RELACIONAMENTOS

Qual o nome do RELACIONAMENTO ?

Figura E.21: Modelando relacionamento entre classes.

Caso o relacionamento seja aceito pelo sistema, será mostrada uma tela (Fig. E.22) para obtenção das cardinalidades mínima e máxima.

MODELAGEM DE RELACIONAMENTOS

CLASSE ORIGEM : MODELO
 CLASSE DESTINO: FABRICANTE
 Existem no MINIMO quantos(as)
 MODELO
 para cada
 FABRICANTE Valor:

Figura E.22: Obtendo cardinalidade.

Em seguida, será apresentada uma tela (Fig. E.23) onde será questionada a necessidade de se manterem valores anteriores

para o relacionamento.

MODELAGEM DE RELACIONAMENTOS

CLASSE ORIGEM : MODELO
CLASSE DESTINO: FABRICANTE
Todo(a) ou Todos(as)
MODELO
Esta(ao) relacionado(a) com APENAS UM(A)
FABRICANTE
Confirme S/N:::)

Figura E.23: Preservando valores anteriores.

Em seguida, o usuário deverá confirmar se o relacionamento é chave, é parte da chave ou não pertence à chave da classe origem.

E.3.2.5 Relacionamento TEMPORAL.

É apresentada uma tela (Fig. E.24) na qual o usuário fornecerá o nome da classe que dá origem ao relacionamento.

Caso a classe seja aceita pelo sistema, é apresentada uma tela para obtenção do nome da classe que é destino do relacionamento.

Sendo a classe aceita pelo sistema, o usuário deverá confirmar se toda classe origem deve obrigatoriamente ir, ou poderá ir para a classe destino.

Em seguida o usuário deverá confirmar se toda a classe

destino deve obrigatoriamente vir da classe origem ou se uma classe destino pode vir da classe origem.

MODELAGEM DE RELACIONAMENTOS TEMPORAIS

Qual o nome da CLASSE que dá origem ao relacionamento?

Figura E.24 : Modelando relacionamentos temporais.

E.3.3 Modelagem de Tempo.

É apresentada na tela (Fig. E.25) onde o sistema pergunta se o usuário deseja associar parâmetro de tempo à uma determinada classe.

MODELAGEM DE TEMPO

Deseja associar parametro de tempo 'a CLASSE

CAIXA (S/N) ?

Figura E.25 : Associando tempo às classes.

Em caso afirmativo é apresentada uma tela (Fig. E.26) para obtenção do parâmetro de tempo da classe.

Caso o tempo de existência seja quantificado com o valor 0 (zero) o parâmetro de tempo da classe é excluído (Fig. E.27).

MODELAGEM DE TEMPO

Informe parametro de tempo da classe
CAR_DE_FABR

tempo de existencia: 3
unidade de tempo: ano

Figura E.26 : Especificando tempo localizado ou quantidade de tempo.

MODELAGEM DE TEMPO

Informe parametro de tempo da classe
CAR_EM_USO

tempo de existencia: 0
unidade de tempo:

Confirme a exclusão do parametro de tempo desta classe: (S/N)

Figura E.27 : Alteração do parâmetro de tempo.

E.3.4 Relatório das TECDs.

É responsável pela impressão das tabelas das TECDs e pela geração dos arquivos (*.map) que são as TECDs padronizadas ao posterior mapeamento, para um esquema interno relacional.

Neste módulo, serão feitos relatórios da situação das tabelas de CLASSE (Fig. E.28), AGRUPAMENTOS (Fig. E.29), AGREGAÇÃO (Fig. E.30), RELACIONAMENTOS TEMPORAIS (Fig. E.31), GENERALIZAÇÃO-PAPÉIS (Fig. E.32) e RELACIONAMENTOS (Fig. E.33).

Antes de imprimir cada tabela, o sistema solicita confirmação sobre o posicionamento do papel na impressora.

TABELA DE CLASSES

NOVE DA CLASSE	TIPO	TIPO	TIPO	TIPO	TIPO	TIPO	TIPO
	CLASSE	ENTIDADE	TEMPO	DE VIDA	UNIDADE	EVENTO	DELETADA
CARRO	C	N	N	000		N	N
FABRICANTE	C	K	N	000		N	N
CAR_DE_FABR	C	N	S	003	ano	N	N
CAR_EX_GARAGEM	C	N	S	002	ano	N	N
CAR_EX_USO	C	N	S	003	ano	N	N
PESSOA	C	K	N	000		N	N
GRP_PROPRIETARIO	C	N	N	000		N	N
CAR_DESTRUIDO	C	K	S	003	ano	N	N
DATA	C	N	N	000		N	N
MUM_REGISTRO	C	N	N	000		N	N
MUM_SERIE	C	N	N	000		N	N
NOME_MODELO	C	K	N	000		N	N
NOME_FABR	D	S	N	000		N	N
MODELO	C	K	N	000		N	N
CONS_COMBUSTIVEL	C	N	N	000		N	N
QUANTIDADE	S	I	N	000		N	N
ANO_PRODUCAO	C	N	N	000		N	N
FABR_POR_ANO	C	N	N	000		N	N
QUANT_CARRO	C	N	N	000		N	N
GARAGEM	C	K	N	000		N	N

Figura E.28 : Relatório da tabela de classes.

TABELA DE AGRUPAMENTOS		
CLASSE ELEMENTO	CLASSE UNICO	TIPO AGRUP
PESSOA	GRUPO_PROPRIETARIO	A

Figura E.29 : Relatório da tabela de agrupamentos.

TABELA DE AGREGACAO		
CLASSE COMPONENTE	CLASSE AGREGADA	TIPO AGRREG
ANO_PRODUCAO	FABR_POR_ANO	A
FABRICANTE	FABR_POR_ANO	A

Figura E.30 : Relatório da tabela de agregações

TABELA DE RELACIONAMENTOS TEMPORAIS PRÉ-POÓS		
PRÉ CLASSE	POÓS CLASSE	TIPO RELAC (PRE/POÓS)
CAR_DE_FABR	CAR_EX_GARAGEM	es
CAR_EX_GARAGEM	CAR_EX_USO	ee
CAR_EX_USO	CAR_EX_GARAGEM	ss
CAR_EX_USO	CAR_DESTRUÍDO	se

Figura E.31 : Relatório da tabela de relacionamentos temporais pré-pós.

TABELA DE GENERALIZACAO - PAPEIS			
SUB CLASSE	CLASSE GENERALIZADA	PAPEL	TIPO RELACAO (GENERALIZACAO)
CAR_DE_FABR	CARRO	situacao propriedade	E 0
CAR_EX_GARAGEM	CARRO	situacao propriedade	E 0
CAR_EX_USO	CARRO	situacao propriedade	E 0
CAR_DESTRUÍDO	CARRO	situacao propriedade	E 0

Figura E.32 : Relatório da tabela de generalização-papéis.

TABELA DE RELACIONAMENTOS

CLASSE ORIGEM	RELACIONAMENTO	CLASSE DESTINO	ICARDI	ICARDI	IDENT	PK	OR	QUANT	TIPO	PK	OR	QUANT	TIPO	ICARDI	ICARDI	IDENT	PK	OR	QUANT	TIPO	PK	OR	QUANT	TIPO
CARRO	tem_mua_res	NUM_REGISTRO	001	001	1			0				0												
NUM_REGISTRO	inv_tem_mua_res	CARRO	001	001	1			0				0												
CARRO	es_do_modelo	MODELO	001	001	0			0				0												
MODELO	es_modelo_de	CARRO	000	0	0			0				0												
CARRO	produzido_em	ANO_PRODUCAO	001	001	2			0				0												
ANO_PRODUCAO	inv_produzido_em	CARRO	000	0	2			0				0												
CARRO	produzido_por	FABRICANTE	001	001	0			0				0												
FABRICANTE	pr_mua	CARRO	001	0	0			0				0												
CARRO	tem_mua_ser	NUM_SERIE	001	001	2			0				0												
NUM_SERIE	inv_tem_mua_ser	CARRO	001	001	2			0				0												
FABRICANTE	es_nome_de	NOME_FABR	001	001	1			0				0												
FABRICANTE	e_proprietario	CAR_DE_FABR	001	0	0			0				0												
CAR_DE_FABR	e_proprietario	FABRICANTE	001	001	0			0				0												
FABRICANTE	fabrica	MODELO	001	0	0			0				0												
MODELO	tem_fabr	FABRICANTE	001	001	0			0				0												
FABRICANTE	quant_fabr	QUANTIDADE	001	001	0			0				0												
QUANTIDADE	inv_quant_fabr	FABRICANTE	001	001	0			0				0												
CAR_EX_GARAGEM	es_proprietario	GARAGES	001	001	0			0				0												
GARAGES	es_proprietario	CAR_EX_GARAGEM	001	0	0			0				0												
CAR_EX_USO	es_proprietario	PESSOA	001	0	0			2				0												
PESSOA	es_proprietario	CAR_EX_USO	001	0	0			2				0												
CAR_EX_USO	es_do_grupo	GRP_PROPRIETARIO	001	001	0			2				0												
GRP_PROPRIETARIO	grp_pessoa	CAR_EX_USO	001	0	0			2				0												
CAR_DESTRUIDO	data_destr	DATA	001	001	0			0				0												
DATA	inv_data_destr	CAR_DESTRUIDO	000	0	0			0				0												
MODELO	tem_nome	NOME_MODELO	001	001	1			0				0												
NOME_MODELO	inv_tem_nome	MODELO	001	001	1			0				0												
MODELO	tem_cons_comb	CONS_COMBUSTIVEL	001	001	0			0				0												
CONS_COMBUSTIVEL	do_modelo	MODELO	000	0	0			0				0												
FABR_POR_ANO	cons_med_comb	CONS_COMBUSTIVEL	001	001	0			0				0												
CONS_COMBUSTIVEL	inv_cons_med_comb	FABR_POR_ANO	000	0	0			0				0												
FABR_POR_ANO	quant_car_reg	QUANT_CARRO	001	001	0			0				0												
QUANT_CARRO	inv_quant_car_reg	FABR_POR_ANO	000	0	0			0				0												
FABRICANTE	cons_med_comb	CONS_COMBUSTIVEL	001	001	0			0				0												
CONS_COMBUSTIVEL	cons_med_comb	FABRICANTE	000	0	0			0				0												

Figura E.33 : Relatório da tabela de relacionamentos.

E.3.5 Finaliza modelagem.

Será impresso um quadro (Fig. E.34) demonstrando a situação da modelagem, informando, para cada classe, o que foi modelado e o que falta modelar.

CLASSE	TEMPO EVENTO DELETADA	NOME_REAC CLAS_ORIGEM CLAS_DESTINO IDENT_CHAVE	CARD_MIN CARD_MAX VAL_ANTER QTE_VALOR	TIPO_REL DELETADO PRE_CLAS POS_CLAS	RELPREPOS SUBCLASSE CLAS_GEMER NOME_PAPEL	TIP_PAPEL REL_CONJUNTO CLAS_COMPOSITE CLAS_AGREGADA	TIP_AGREGACAO CLAS_ELEMENTO CLAS_AGRUPADA TIP_AGRUPADO
CARRO	FK FK FK FK	ND ND ND ND	ND ND ND ND	ND ND ND ND			
FABRICANTE	FK FK FK FK	ND ND ND ND	ND ND ND ND	ND ND ND ND			
CAR_DE_FABR	FK FK FK FK	ND ND ND ND	ND ND ND ND	ND ND ND ND			
CAR_EM_GARAGEM	FK FK FK FK	ND ND ND ND	ND ND ND ND	ND ND ND ND			
CAR_EM_USO	FK FK FK FK	ND ND ND ND	ND ND ND ND	ND ND ND ND			
PESSOA	FK FK FK FK	ND ND ND ND	ND ND ND ND	ND ND ND ND			
GRP_PROPRIETARIO	FK FK FK FK	ND ND ND ND	ND ND ND ND	ND ND ND ND			

* FK: falta modelar.
* ND: modelado.

** SISTEMA DESCONT **

Figura E.34 : Relatório de diagnóstico das classes modeladas.

APÊNDICE F

MENSAGENS EXIBIDAS PELO SISTEMA DESCONT-THM

Durante a execução do DESCONT-THM serão exibidas várias mensagens referentes a uma situação corrente. Apresentamos neste apêndice o conjunto de mensagens e sua procedência, com objetivo de auxiliar o usuário à execução do sistema.

MENSAGEM 1 : Geração cancelada - Arquivo já existe.

Esta mensagem será exibida na geração de um arquivo quando for solicitada na função de manutenção do BC-THM; ocorre se um dos arquivos escolhidos no mapa geral de arquivos já tiver sido gerado.

MENSAGEM 2 : Inclusão rejeitada - Senha já existe.

Será mostrada esta mensagem durante a execução da função de geração de senhas, onde a senha informada ao sistema já existe catalogado no arquivo "SENHA.DAT".

MENSAGEM 3 : Geração realizada com sucesso.

Embora esta mensagem não represente um erro, será exibida em caso de sucesso tanto na geração de arquivos de dados quanto na geração de senhas.

MENSAGEM 4 : Tamanho máximo excedido na
identificação da classe.

Este erro será detectado quando for atribuído a uma classe,
um string com tamanho superior a 25 caracteres.

MENSAGEM 5 : Caractere inválido na
identificação da classe.

Será exibida esta mensagem quando for especificado ao nome
da classe algum dos caracteres especiais do tipo
"~, ', !, @, #, \$, %, ^, &, *, (,), -, +, =, /, ?, >, <, ;, :, {, }, [,]"

MENSAGEM 6 : Classe já existe.

Esta mensagem será exibida em caso de determinação de um
nome que identificará uma classe e esta classe já tenha sido
modelada. (Existindo no arquivo CLASSE.DAT).

MENSAGEM 7 : Identificação da classe não foi considerada.

A rejeição da identificação da classe deve-se ao não
estanciamiento de um valor a uma classe que se está tentando
modelar.

MENSAGEM 8 : Classe origem não pode ser
igual à classe destino.

Este erro será detectado durante a modelagem de

relacionamentos entre classes, quando for atribuído o mesmo nome para as classes origem e classe destino.

MENSAGEM 9 : Relacionamento só pode ser modelado como hierárquico É-UM, para determinar uma GENERALIZAÇÃO.

Esta mensagem de erro será mostrada durante a modelagem de relacionamentos, quando a classe origem ou destino tiver sido modelada anteriormente como classe GENERALIZADA, não permitindo estabelecer relacionamento diferente de hierárquica do tipo É-UM.

MENSAGEM 10 : Relacionamento só pode ser modelado como hierárquico É-PARTE para determinar uma AGREGAÇÃO.

Este erro ocorre quando a modelagem da classe origem ou destino tiver sido modelada anteriormente como classe AGREGADA, não permitindo modelagem diferente de hierárquico do tipo É-PARTE.

MENSAGEM 11 : Relacionamento só pode ser modelado como hierárquico É-ELEMENTO para determinar um AGRUPAMENTO.

Este erro ocorre durante a modelagem de relacionamento quando a classe origem ou destino tiver sido modelada anteriormente como classe AGRUPADA, não permitindo modelagem

diferente de hierárquico do tipo E-ELEMENTO.

MENSAGEM 12 : Relacionamento só pode ser modelado
como PRÉ-PÓS para determinar
relacionamento TEMPORAL.

Este erro ocorre durante a modelagem de relacionamento quando a classe origem ou destino tiver sido modelada anteriormente como pré-classe em um relacionamento TEMPORAL, não permitindo uma modelagem diferente de temporal hierárquico do tipo PRÉ-PÓS.

MENSAGEM 13 : Tamanho máximo excedido na
identificação do relacionamento.

Será exibida esta mensagem de erro quando for atribuído a um relacionamento, um string com tamanho superior a 25 caracteres.

MENSAGEM 14 : Classe desconhecida.

Este erro ocorre quando se especifica um nome à classe origem ou destino sem que esta tenha sido modelada anteriormente.

MENSAGEM 15 : A classe generalizada não
pode ser uma classe pré-pós.

Esta mensagem será mostrada quando se tentar estabelecer uma classe como generalizada e a mesma tiver sido anteriormente modelada como classe temporal.