

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Este exemplar corresponde a redação final
da tese defendida por Armando Luiz Nicolini
Dalgado e aprovada pela Comissão Julgadora em
12/01/1987
L. P. Magalhães

BANCO DE DADOS NO CONTEXTO DE
PROJETO AUXILIADO POR COMPUTADOR

187

por: Eng. Armando Luiz Nicolini Dalgado
Orientador: Prof. Dr. Léo Pini Magalhães

Tese de mestrado apresentada a Faculdade de
Engenharia Elétrica da Universidade
Estadual de Campinas

JANEIRO DE 1987

UNICAMP
BIBLIOTECA CENTRAL

Este trabalho foi editado utilizando o sistema
de edição de texto em português
do Laboratório de Sistemas/Mini-Microcomputadores
DCA - FEE - UNICAMP.

Dedico este trabalho
à minha família

AGRADECIMENTOS

Aos meus pais, sem os quais dificilmente este trabalho seria realizado.

Ao Professor Dr. Léo Pini Magalhães, pela confiança, incentivo e orientação.

Aos colegas Ivan, Sérgio e Helga, pela colaboração indispensável prestada durante o trabalho.

Aos amigos Ivan, Sérgio, José Raimundo, Lotufo, José Mario, Helga e estagiários do LS/M pelo companheirismo e amizade.

À ITAUTEC, pela importante colaboração nos estudos sobre sistemas PAC.

çA CAPES e à FAPESP, pelo apoio financeiro.

E a todos aqueles que participaram, direta ou indiretamente, na elaboração deste trabalho.

RESUMO

Neste trabalho é proposto um Modelo de Dados para aplicação em Projeto Auxiliado por Computador - PAC - que permite uma modelagem adequada e precisa da Informação ligada a esta aplicação. A motivação deste trabalho deve-se à forte tendência atual de Inclusão de Sistemas de Banco de Dados para o tratamento das complexas informações existentes em um projeto de qualquer natureza e, conseqüentemente, em Sistemas Automatizados para Apoio a Projeto.

O Modelo Entidade-Relacionamento para PAC (MER/PAC) é uma extensão do Modelo Entidade-Relacionamento (MER), levando em consideração dois aspectos que caracterizam aplicações PAC: o aspecto micro, que se refere à arte de projetar; e o aspecto macro, que se refere à sistemática de projeto.

No MER/PAC são introduzidos os conceitos de OBJETO COMPLEXO, ASSOCIAÇÃO COMPLEXA, GENERALIZAÇÃO, VERSÃO e EXTENSÃO. Para isto, são definidas primitivas de modelagem que suportam tais conceitos, sendo mantidas as definições para entidades e relacionamentos originais do MER.

Como exemplo da aplicabilidade do modelo proposto, modela-se um sistema para projeto de circuitos impressos, utilizando as primitivas do MER/PAC. Esta modelagem será feita tanto a nível da descrição estrutural do objeto de projeto (placa de circuito impresso), quanto a nível da descrição dos interrelacionamentos dos dados associados aos processos que constituem o sistema-exemplo.

ABSTRACT

This work proposes a Data Model for Computer Aided Design, - CAD - applications, allowing an appropriate modeling of the related application information. The motivation of this work is the current tendency to include Data Base Systems in the treatment of complex informations related to the design process and, therefore, in Automatic Systems for Design Support.

The Entity-Relationship Model for CAD (ERM/CAD) is an extension of the Entity-Relationship Model (ERM), which considers two characteristic aspects of CAD applications: the micro aspects, which refers to the art of design; and the macro aspect, which refers to the systematics of design.

The ERM/CAD utilizes the concepts of COMPLEX OBJECT, COMPLEX ASSOCIATION, GENERALIZATION, VERSION and EXTENSION. Modeling primitives which support such concepts and keep the definitions for entities and relationships of former ERM are defined.

As an example of the applicability of the proposed model, a printing board design system is modeled, using the ERM/CAD primitives. This modeling will comprise the level of structural description of the design object (mother board) and the level of the data relationship description, associated with the processes that establish the example-system.

ÍNDICE

CAPÍTULO 1. INTRODUÇÃO	1
CAPÍTULO 2. SISTEMAS DE BANCOS DE DADOS	6
2.1. Introdução	6
2.2. Processamento de Dados - Histórico	10
2.3. Características de Sistemas de Banco de Dados	13
2.3.1. Compartilhamento de Dados	13
2.3.1.1. Redundância	14
2.3.1.2. Inconsistência	14
2.3.1.3. Privacidade e Segurança	15
2.3.1.4. Integridade	15
2.3.2. Independência de Dados	16
2.3.3. Desempenho	16
2.3.3.1. Sintonização	17
2.3.3.2. Migração de Dados	17
2.3.4. Integração	18
2.4. Abstração de Dados	19
2.4.1. Nível Conceitual	21
2.4.2. Nível Descritivo	22
2.4.3. Nível Organizacional	27
2.4.4. Nível Físico	29
2.5. Modelos de Dados	31
2.5.1. Modelos de Dados - Rede e Relacional ..	35
2.5.1.1. Modelo Rede	35
2.5.1.2. Modelo Relacional	37

2.5.2. Modelo Entidade-Relacionamento	40
2.5.2.1. Conceitos Gerais	40
2.5.2.2. Estudos Adicionais sobre o MER	45
CAPÍTULO 3. SISTEMAS DE PROJETO AUXILIADO POR COMPUTADOR ...	49
3.1. Introdução	49
3.2. Arquitetura de Sistemas PAC	51
3.2.1. Componentes de um Sistema PAC	53
3.2.1.1. Componentes Funcionais	54
3.2.1.2. Componentes de Hardware	56
3.2.1.3. Componentes de Software	57
3.3. Modelagem de Dados PAC	60
3.3.1. Análise Geral	60
3.3.2. O Aspecto Micro de PAC	63
3.3.3. O Aspecto Macro de PAC	64
3.3.4. Modelagem do Processo de Projeto	65
3.3.5. O Processo de Projeto em Ambiente PAC ..	74
CAPÍTULO 4. MODELO ENTIDADE-RELACIONAMENTO PARA PAC	81
4.1. Introdução	81
4.2. Sistemas de Banco de Dados para PAC	82
4.3. O MER/PAC	90
4.4. Primitivas do MER/PAC	93
4.5. Considerações Finais	106
CAPÍTULO 5. EXEMPLO DE APLICAÇÃO	111
5.1. Introdução	111
5.2. Projeto de Circuito Impresso - Modelagem Conceitual	114

5.2.1. Descrição Geral	114
5.2.2. Modelagem Conceitual	116
5.2.2.1. Grupo A - Regras Referentes a Componentes	117
5.2.2.2. Grupo B - Regras Referentes a Rotas	118
5.2.2.3. Grupo C - Regras Referentes a Componentes, Rotas, Ligações e Sinais	120
5.2.2.4. Estrutura de Dados e Regras de Manipulação	125
5.3. Modelo Descritivo	125
5.4. Modelagem do Processo de Projeto	132
CAPÍTULO 6. CONCLUSÕES	139
CAPÍTULO 7. BIBLIOGRAFIA	146

CAPÍTULO 1. INTRODUÇÃO

Um processo de projeto consiste basicamente no desenvolvimento de um conjunto de informações referentes a um determinado objeto - o objeto de projeto - de forma que, ao final deste processo, estas informações permitam a concretização física do objeto, seja através da fabricação do produto, seja por qualquer outra atividade.

Durante a atividade de projeto, o projetista lida com uma gama muito grande de informações, que vai desde as informações de especificação do objeto até o conjunto completo de informações sobre o produto final, passando por informações técnicas de apoio a projeto (normas, tabelas de materiais, etc.); informações de e para outros processos de projeto; informações sobre a análise de resultados intermediários; etc. Em vista disso, o controle e manipulação das informações de projeto se torna muito complexo, exigindo em alguma parte do processo, de preferência em sua totalidade, alguma forma de automatização, seja do acesso às informações, seja dos métodos de engenharia empregados.

Com a evolução dos sistemas computacionais, foi possível estabelecer uma forma eficaz de automatização para o processo de projeto utilizando tais sistemas. A partir daí surge o concepção de Sistemas de Projeto Auxiliado por Computador, ou Sistemas PAC, que são sistemas que utilizam o computador como ferramenta para o controle das informações de um projeto, isto é, as informações

referentes a um produto a ser projetado e as informações a respeito da evolução e do controle dos processos de projeto que coexistem no sistema.

O fluxo de informação existente entre vários processos de projeto em um sistema PAC, bem como o fluxo de informações necessário para um processo em particular são questões de importância fundamental em PAC. Desta forma, Sistemas de Banco de Dados passam a ter um papel importante como suporte aos Sistemas PAC, tanto pela sua capacidade de gerir eficazmente informações, através do pré-processamento de todas as informações referentes a um determinado objeto de projeto, como por possibilitarem um controle do fluxo de informações entre os diversos processos de projeto que coexistem em um ambiente PAC.

Em Sistemas de Projeto em geral, e em Sistemas PAC em particular, pode-se distinguir dois aspectos:

- * ASPECTO MICRO: corresponde aos aspectos envolvidos na definição e manipulação das informações de um objeto de projeto isolado;
- * ASPECTO MACRO: corresponde aos aspectos decorrentes da forma como o processo de projeto evolui e como os processos existentes em um sistema PAC interagem, impondo, eventualmente, novas restrições aos objetos de projeto.

Cada um destes aspectos lida com informações de características diferentes, tendo assim exigências diferentes com

relação à forma de manipulação (acesso, atualização) das informações.

Todas as características e restrições operacionais do Sistema PAC devem ser descritas com máxima precisão pelos especialistas em projeto (engenheiros de um modo geral), de modo que o Administrador de Banco de Dados (ABD) tenha indicações suficientes para definir precisamente uma estrutura de dados compatível com as exigências da Aplicação. A formalização desta descrição, através de um MODELO DE DADOS, é uma ferramenta poderosa para evitar a ambiguidade na interpretação dos requisitos a serem atendidos pelo Sistema de Banco de Dados.

O presente trabalho tem por objetivo apresentar um Modelo de Dados aplicável a PAC, através da definição de primitivas de modelagem que permitam a formalização de todos os requisitos impostos pelo Sistema PAC. O modelo proposto é denominado MODELO ENTIDADE-RELACIONAMENTO PARA PAC (MER/PAC - /DELGA 85/, /DELGA 86a/, /DELGA 86b/). A estrutura deste modelo baseia-se essencialmente nos conceitos já definidos no Modelo Entidade-Relacionamento proposto em /CHEN 76/, com extensões e modificações apropriadas ao tratamento dos aspectos micro e macro que caracterizam um processo de projeto PAC. Entre os modelos mais difundidos e para os quais já se encontram implementações ou propostas de implementações (Rede, Hierárquico, Relacional, Entity-Set), escolheu-se o Modelo Entidade-Relacionamento como modelo-base por apresentar uma semântica mais adequada para a descrição de aplicações PAC.

Resumidamente, este trabalho aborda os seguintes tópicos:

No Capítulo 2 apresenta-se uma visão geral de Sistemas de Banco de Dados, dando-se ênfase aos seus diferentes níveis de abstração, de forma a permitir uma maior independência física e lógica dos dados. São discutidos, sob o ponto de vista da descrição formal do Mundo Real, três Modelos de Dados: Rede, Relacional e Entidade-Relacionamento.

No Capítulo 3 é feita uma abordagem geral dos princípios básicos de Sistemas de Projeto Auxiliado por Computador (arquitetura geral e controle do processo de projeto), dando-se ênfase à natureza e ao volume de informações necessárias para a evolução de um processo de projeto dentro de um Sistema PAC. A finalidade é mostrar algumas características particulares de aplicações PAC que são difíceis de serem representadas de forma natural pelos Modelos de Dados existentes.

No Capítulo 4 apresenta-se uma proposta de um Modelo de Dados para descrição formal das informações envolvidas num Sistema PAC - Modelo Entidade-Relacionamento para PAC. Este modelo é descrito sob o ponto de vista conceitual, abordando-se detalhadamente as suas primitivas.

No Capítulo 5 é apresentado um exemplo de aplicação, onde o Sistema de Banco de Dados é utilizado para gerir as informações necessárias ao Sistema para Projeto de Circuitos Impressos. Através deste exemplo, é ilustrada a aplicabilidade do MER/PAC para a representação de todas as informações semânticas identificadas para

um objeto de projeto, no caso, placas de circuito impresso. O MER/PAG é usado, também neste exemplo, para a representação das informações pertinentes aos processos constituintes do sistema e ao interrelacionamento entre eles.

No Capítulo 6 são apresentadas as conclusões do trabalho, seguidas de sugestões e comentários das diversas possíveis linhas de continuidade para o presente trabalho.

CAPÍTULO 2. SISTEMAS DE BANCO DE DADOS

2.1. Introdução

Um SISTEMA DE BANCO DE DADOS, ou simplesmente BANCO DE DADOS, consiste em uma coleção de dados interrelacionados que são armazenados, sendo que o armazenamento e o acesso a tais dados são controlados por um pacote de programas pertencentes ao sistema e que se encarrega de tornar a utilização do Sistema uniforme à aplicação.

O conjunto de dados a partir dos quais serão derivadas as informações necessárias aos programas de aplicação é denominado BASE DE DADOS (/ULLM 80/) ou DADOS OPERACIONAIS (/DATE 84/). O software que controla o acesso à Base de Dados denomina-se o SISTEMA DE GERÊNCIA DE BANCO DE DADOS (SGBD). Existe ainda uma equipe de pessoas conhecida como o ADMINISTRADOR DE BANCO DE DADOS (ABD), que se responsabiliza pelo mapeamento das informações definidas pelos usuários de aplicação às primitivas do modelo de dados suportado pelo SGBD disponível. Muitas vezes, o ABD é considerado como um membro integrante do Sistema de Banco de Dados (/DATE 84/), devido ao fato de que o desempenho de um Banco de Dados depende das decisões sobre as estratégias de acesso e estruturação física/lógica tomadas pelo ABD.

O SISTEMA DE GERÊNCIA DE BANCOS DE DADOS (SGBD) é formado por um conjunto de procedimentos que isola os programas de aplicação dos detalhes referentes à criação, busca, armazenamento,

modificação, segurança e estruturação física da Base de Dados (/TSIC 77/). Cada usuário deve poder acessar qualquer grupo de dados a que ele tenha direito, sem ter de se preocupar com a real organização e estruturação interna dos mesmos.

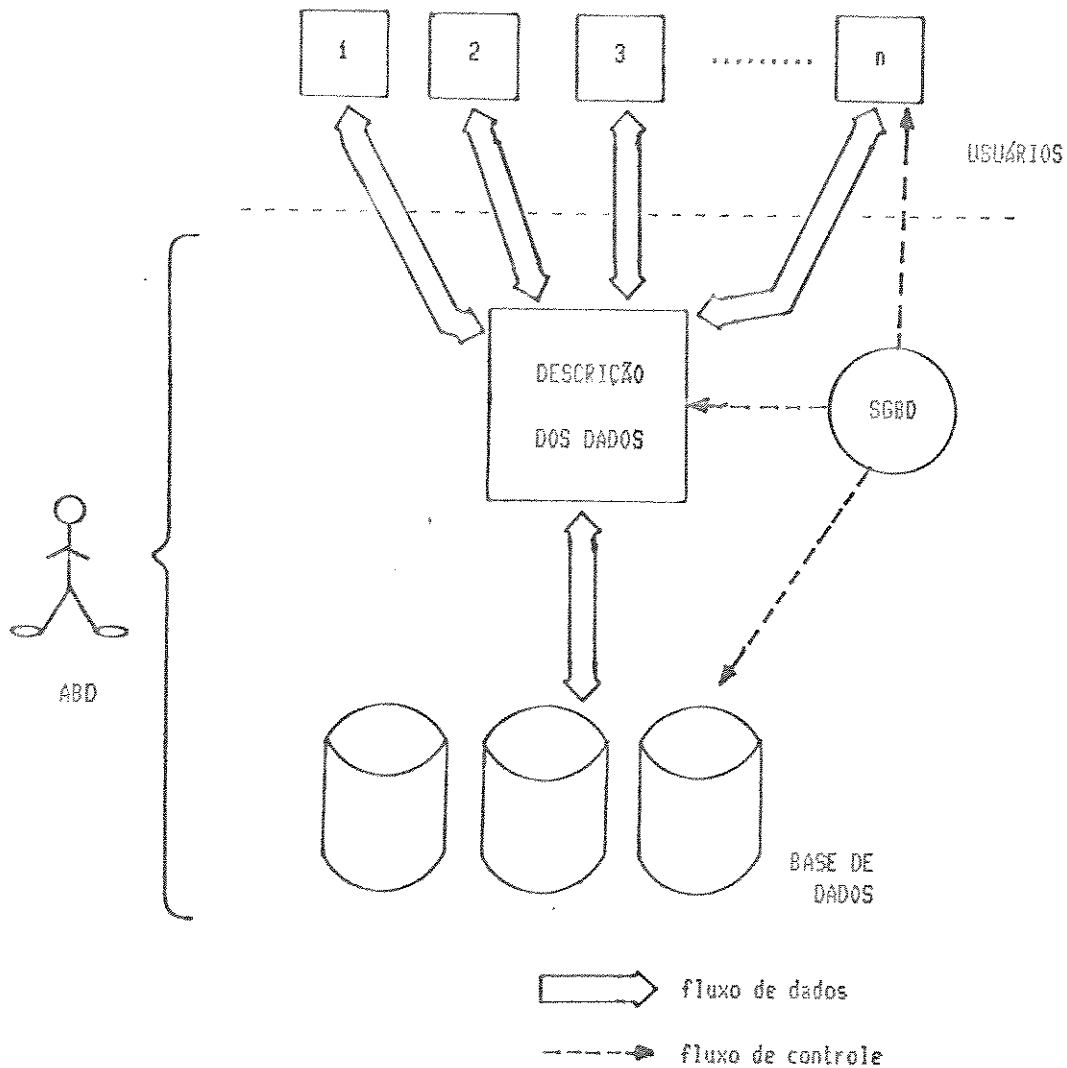


Figura 2.1. Sistema de Banco de Dados

Assim, um SGBD deve incluir tanto funções que facilitem a definição das estratégias de estruturação dos dados físicos/lógicos por parte do ABD, como também os acessos por parte dos usuários

(programas de aplicação) (figura 2.1). Ao longo deste texto, serão usados indistintamente os termos usuários e programas de aplicação.

Para melhor caracterizar as funções de um SGBD, serão abordadas, em maior detalhe, dois objetivos de um Sistema de Bancos de Dados:

- * Independência dos dados;
- * compartilhamento dos dados.

A INDEPENDÊNCIA DOS DADOS refere-se ao isolamento entre o modo de utilização dos dados, por parte dos usuários, e a estrutura de armazenamento e estratégia de acesso dos mesmos. Com isso, objetiva-se a redução da interdependência entre os programas de aplicação e a forma de tratamento dos dados requisitados. O isolamento faz-se com a definição de uma interface, que mapeia a representação lógica na representação física dos dados e vice-versa. Conforme a complexidade e a potencialidade das funções incluídas no SGBD, distinguem-se diferentes graus de independência: INDEPENDÊNCIA SIMPLEMENTE FÍSICA, INDEPENDÊNCIA FÍSICA E PARCIALMENTE LÓGICA e INDEPENDÊNCIA FÍSICA E LÓGICA. De modo geral, quanto maior for o grau desta independência, menor será o custo de adaptação dos programas já existentes, caso sejam modificadas as representações internas no computador.

Paralelamente ao princípio de independência de dados surgiu o conceito de ABSTRAÇÃO DE DADOS, que enfatiza não só a representatividade como também a semântica dos dados armazenados (/HAMM 76/). Um conjunto de dados numa Base de Dados pode ser

representado em níveis de abstração semanticamente distintos, cada qual apresentando um diferente grau de independência lógica e/ou física. Quanto maior for o nível de abstração dos dados, maior será a aproximação da semântica destes dados ao Mundo Real visualizado pelos usuários. Este aspecto será abordado em maior profundidade na seção 2.4.

O COMPARTILHAMENTO DOS DADOS é fator importante em aplicações onde o volume de dados é grande e há forte intercâmbio de informações entre os programas que representam a aplicação (figura 2.2). Tal compartilhamento traz como consequência a diminuição da redundância física dos dados, facilitando o controle de consistência. O compartilhamento cria, porém, problemas tais como: a garantia da privacidade dos usuários em relação aos seus dados particulares; a possibilidade de ocorrência de duas ou mais operações simultâneas e incompatíveis, levando a intertravamento ("deadlock"), se a coordenação do sistema falhar; etc. Assim, um SGBD deve ser provido de funções para verificação de autorização e de procedimentos de validação.

Como os programas de aplicação que compartilham os mesmos dados não precisam apresentar finalidades necessariamente semelhantes, o SGBD deve ainda prover recursos que abstraiam diferentes VISÕES do Mundo, a partir de uma mesma Base de Dados. Assim, para cada usuário, é como se os dados fossem estruturados para o seu uso exclusivo.

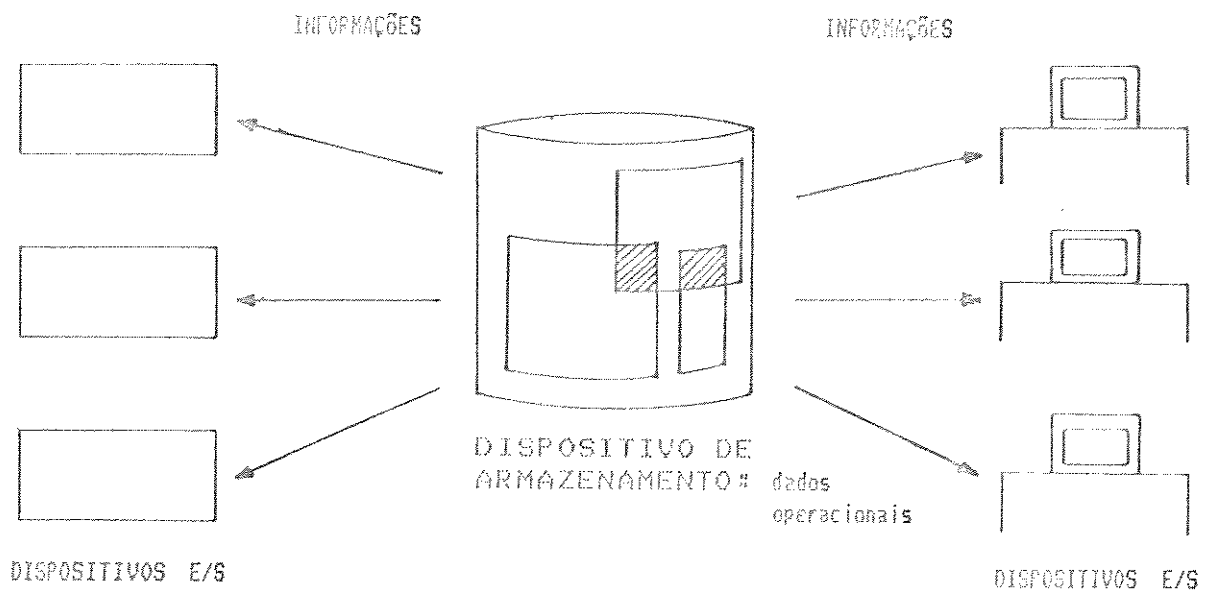


Figura 2.2. Compartilhamento de dados

2.2. Processamento de Dados - Histórico

Tradicionalmente, programas caracterizam-se por ter uma forte dependência com a forma como seus dados estão estruturados. Programas, mesmo quando utilizam dados semelhantes, demandam arquivos distintos. Enquanto o computador é usado como máquina de calcular, isso não se configura como um problema crítico. Contudo, tal aspecto torna-se indesejável à medida que aumenta a complexidade e a quantidade dos dados dos programas de aplicação. A solução é estabelecer uma Base de Dados comum e gerenciá-la por um sistema de software.

No decorrer da evolução das técnicas de gerência de dados, distinguem-se três fases marcadas, ou pelos avanços tecnológicos na área de hardware ou pela evolução na área de software:

- * primeira fase: programas totalmente dependentes da estrutura física dos dados;
- * segunda fase: Introdzem-se programas dependentes da organização lógica dos dados;
- * tercelra fase: Introdzem-se programas independentes da organização lógica dos dados.

Os programas totalmente dependentes da estrutura física exigem que o usuário desenvolva não só o software de aplicação como também, o software de estruturação de dados e operações de E/S. A estrutura de dados é totalmente dependente de uma aplicação específica. Quando ocorrem modificações na organização dos dados e/ou nos dispositivos de armazenamento, os programas de aplicação devem ser alterados, recompilados e retestados devidamente. Muitas vezes, diferentes programas de aplicação que necessitam dos mesmos dados, porém sob formas diferentes, demandam arquivos distintos.

Programas somente dependentes da organização lógica dos dados caracterizam-se pelo isolamento entre programas de aplicação e estrutura física dos arquivos, através de um pacote de software, incluído no próprio Sistema Operacional, que mascara, através de uma Interface Base de Dados/Programas de Aplicação, todos os detalhes da organização física dos dados, tornando os programas independentes dos dispositivos de armazenamento.

A grande difusão do uso de computadores em todas as áreas causou um aumento no interesse de isolamento entre programas e sua estrutura lógica de dados (por ex.: tabelas de acesso, listas de

células lógicas ligadas para guardar informações de tamanho variável e imprevisível, etc). Desta forma, os programas manter-se-iam inalterados, mesmo que ocorressem modificações nos endereços e nas estratégias de acesso lógico aos dados, pois a complexidade da estrutura lógica dos dados requeridos pelos diferentes usuários fica transparente a eles. Sob o ponto de vista destes usuários, os dados estariam estruturados numa forma simples, estrutura esta definida através de uma ferramenta formal e precisa, denominada Modelo de Dados. Para isso, surgiu o conceito de SISTEMA DE BANCO DE DADOS, que provê recursos para mapear a estrutura definida através de um Modelo de Dados na estrutura de dados lógicos e físicos implementada no sistema computacional.

Modelos de Dados constituem um meio eficiente de comunicação entre usuários e Sistemas de Banco de Dados. Nos últimos anos, dezenas de Modelos de Dados foram propostos, objetivando encontrar um Modelo que apresente primitivas de modelagem e regras bem definidas de sintaxe e semântica, para descrever sem ambiguidade as percepções dos usuários sobre o Mundo Real que os envolve. Classificam-se estes modelos em duas grandes classes:

- * os direcionados à concepção computacional;
- * os direcionados à concepção do Mundo Real.

Sob o ponto de vista de implementação, os SGBD's comercialmente disponíveis hoje em dia suportam Modelos de Dados mais próximos à concepção computacional. Pesquisas e estudos têm sido feitos no sentido de estender estes SGBD's através da inclusão de funções que estabeleçam a correspondência biunívoca entre as

primitivas definidas nestas duas diferentes percepções, automatizando assim a transformação da estrutura de informações familiar aos usuários em estrutura de dados suportada pelo computador.

2.3. Características de Sistemas de Banco de Dados

Conforme definido anteriormente (Item 2.1), os principais objetivos, a serem atendidos com o uso de um Sistema de Banco de Dados, podem ser resumidos basicamente em: proporcionar INDEPENDÊNCIA e COMPARTILHAMENTO DE DADOS, almejando uma MELHOR UTILIZAÇÃO dos mesmos e INTEGRANDO-os como um conjunto global. Em seguida, abordar-se-á mais detalhadamente cada um destes aspectos.

2.3.1. Compartilhamento de Dados

Com um sistema que permita o compartilhamento dos dados, o volume de dados redundantes pode ser reduzido. Isto, embora signifique uma redução do custo por bit de armazenamento, torna ao mesmo tempo necessário um conjunto de algoritmos complexo e extenso para controlar o acesso aos mesmos dados operacionais pelos diferentes programas de aplicação.

O compartilhamento de dados permite não só a redução do volume do espaço ocupado (redução de dados redundantes) como também possibilita a utilização dos dados já existentes por novos programas de aplicação. Ligado ao compartilhamento de dados, devem-se ressaltar os aspectos relativos ao controle de redundância,

Inconsistência, segurança, privacidade e integridade de dados.

2.3.1.1. Redundância

Dados redundantes podem ocupar desnecessariamente espaços físicos e encarecer as operações de manipulação. Por exemplo, se for atualizado um dado, todos os outros redundantes devem ser alterados adequadamente, para evitar inconsistência. Porém, nem sempre a redundância é indesejável. Na realidade, em muitos casos, recomenda-se a redundância para melhorar o tempo de acesso e simplificar os métodos de endereçamento. Em algumas situações, dados são duplicados fisicamente para assegurar a recuperação destes, caso ocorram falhas acidentais. Portanto, um Sistema de Banco de Dados deve ser capaz de eliminar os dados redundantes desnecessários, para obter um ponto ótimo entre a eficiência do Sistema e o volume de dados redundantes.

2.3.1.2. Inconsistência

Uma base de dados é dita CONSISTENTE num instante t , quando quaisquer grupos idênticos de dados geram resultados de valores iguais para inferências idênticas, satisfazendo ao mesmo tempo as restrições impostas. Note-se que, em certos casos, a inconsistência é uma propriedade intrínseca do sistema que contém os dados redundantes, pois raramente eles são atualizados simultaneamente. A sequência de procedimentos que transfere a Base de Dados de um estado consistente a um novo estado consistente é denominada TRANSAÇÃO. Durante o processo de transformação de estado, a Base de Dados fica "temporariamente" em estado inconsistente (/NEUM 82/). Um

Sistema de Banco de Dados deve prover controles adequados para distinguir uma inconsistência temporária de uma permanente e tomar as ações corretivas necessárias.

2.3.1.3. Privacidade e Segurança

São características mais ligadas aos fatores sociais que aos computacionais propriamente ditos.

A PRIVACIDADE refere-se ao direito que o usuário tem para determinar quando, como e quanto das suas informações privadas podem ser acessadas pelos outros que utilizem a mesma Base de Dados. SEGURANÇA significa a proteção contra alterações, destruição acidental e/ou não autorizada.

2.3.1.4. Integridade

Entende-se como INTEGRIDADE, a manutenção de dados verdadeiros nos arquivos, satisfazendo sempre as restrições impostas pelo Sistema e/ou usuários. Em quaisquer tipos de acidentes (falhas de hardware ou pessoal), o Sistema deve prover meios para recuperar os dados atingidos, garantindo a manutenção da integridade. Alguns recursos utilizados para este fim são: log-file, imagem da sessão, etc. (/DATE B3/).

2.3.2. Independência de Dados

Um Sistema de Banco de Dados deve prover basicamente duas visões independentes de dados - representação física de dados e representação lógica de dados - de forma a garantir que modificações na estrutura física dos dados não afetem os programas de aplicação existentes e vice-versa. A importância desta característica já foi ressaltada no item 2.1.

2.3.3. Desempenho

O desempenho de um Sistema de Banco de Dados está intimamente ligado à organização dos dados (física e lógica), envolvendo como parâmetros de avaliação, o volume do espaço físico ocupado, o acesso aos dados e os tipos de interação usuário/Sistema disponíveis. Por exemplo, para os usuários on-line, o tempo de diálogo homem-máquina (intervalo de tempo entre a entrada do comando e a saída de mensagem) é um fator essencial na avaliação do desempenho do sistema.

Até pouco tempo atrás, só existiam Sistemas de Banco de Dados não inferenciais, o que permitia que o Sistema fosse organizado com certo desempenho a priori. Entretanto, como a tendência atual é a implementação de sistemas inferenciais, estes devem prover recursos adequados para seleccionar melhores caminhos de busca em cada inferência não prevista durante o projeto, assegurando, sempre que possível, o melhor desempenho.

Para melhorar o desempenho do Sistema, recursos como SINTONIZAÇÃO e MIGRAÇÃO DOS DADOS podem ser incorporados entre as facilidades oferecidas pelo SGBD.

2.3.3.1. Sintonização

A medida que os dados sofrem alterações, a sua estrutura pode também sofrer modificações (por exemplo, caminhos de acesso, quantidade de registros, índices, apontadores, etc). Sabe-se que o desempenho dos programas de aplicação depende da estrutura dos dados - sempre existe uma estrutura que é a mais adequada para uma aplicação específica. Portanto, é desejável que o Sistema de Gerência de Banco de Dados possa organizar os dados dos programas de aplicação em uma das estruturas disponíveis mais adequadas, assegurando sempre um bom desempenho. Esta propriedade é importante para os sistemas inferenciais. Nos sistemas existentes, estes ajustamentos (escolha de uma estrutura lógica mais adequada) são feitos baseados em dados estatísticos.

2.3.3.2. Migração de Dados

Em certos sistemas, é interessante agrupar os dados conforme a sua frequência de uso. Uma das estratégias utilizadas é armazenar os dados obsoletos em memórias lentas (fitas, por exemplo) e os de consulta frequente em memórias rápidas (por exemplo, discos). Periodicamente, são feitas verificações de todos os dados existentes na Base de Dados e reorganizadas suas localidades. Este remanejamento de dados é conhecido como MIGRAÇÃO DE DADOS. Deve-se

ressaltar que a redução do tempo de acesso aos dados não é devida apenas à migração física dos dados, mas também definição (ou redefinição) de estratégias de acesso (tabelas hash, novas árvores de busca, etc) que são associadas a estes dados.

2.3.4. Integração

Entende-se como INTEGRAÇÃO das Bases de Dados relacionadas a uma aplicação a administração de todas as informações contidas num Sistema de Banco de Dados como um conjunto global. Cada vez mais, esta é uma necessidade imperiosa das aplicações, trazendo vantagens tais como: menor redundância, maior controle de integridade, maior compartilhamento de dados, melhor utilização de recursos disponíveis, etc. Existem, porém, alguns problemas que devem ser levados em consideração:

- * maior vulnerabilidade à destruição total de dados, se ocorrerem falhas;
- * maior susceptibilidade à quebra de sigilo dos dados dos usuários que compartilhem a mesma Base de Dados;
- * complexidade do software para gerenciar um grande volume de dados destinados às aplicações que podem ser de natureza antagônica, como por exemplo, aplicações em tempo real e aplicações altamente confidenciais.

Face a estes problemas, o grau de integração dos dados, a ser adotado por uma instituição que pretende implementar um Sistema

de Banco de Dados, dependerá da natureza das informações, das aplicações em análise, da disponibilidade do software (por exemplo, o processamento extra necessário ao controle dos problemas citados) e hardware (por exemplo: mecanismos de acesso físico aos dados).

2.4. Abstração de Dados

Para que todas as informações relevantes a uma aplicação possam ser armazenadas num Sistema de Banco de Dados, elas devem ser identificadas e modeladas através de primitivas aceitáveis pelo SGBD para então serem convertidas em dados internos compatíveis com a estrutura física de armazenamento.

Este processo de identificação e modelagem da informação é fruto das diferentes percepções possíveis da informação, que podem ser classificadas em diferentes NÍVEIS DE ABSTRAÇÃO DE DADOS. Embora não haja uma padronização na classificação destes níveis, serão aqui considerados quatro níveis, semelhantes àqueles encontrados em /CHEN 76/, /SETZ 86/ e /WGIO 77/ (figura 2.3):

- * Nível Conceitual;
- * Nível Descritivo;
- * Nível Organizacional:
 - . independente dos caminhos de acesso;
 - . dependente dos caminhos de acesso;
- * Nível Físico.

Inicialmente, acima destes quatro níveis de abstração tem-se o conjunto de informações que de maneira abstrata descreve o

ambiente no qual se insere uma determinada aplicação. Tais informações se encontram disponíveis a nível de idéias, nebulosas sob um ponto de vista formal, constituindo o que se chama Mundo Real. Os elementos do Mundo real são os seres, os fatos, as coisas e os organismos sociais, cabendo ao usuário ou ao projetista do banco de dados definir o que lhe interessa como mundo real para fins de tratamento de informações (/SETZ 86/).

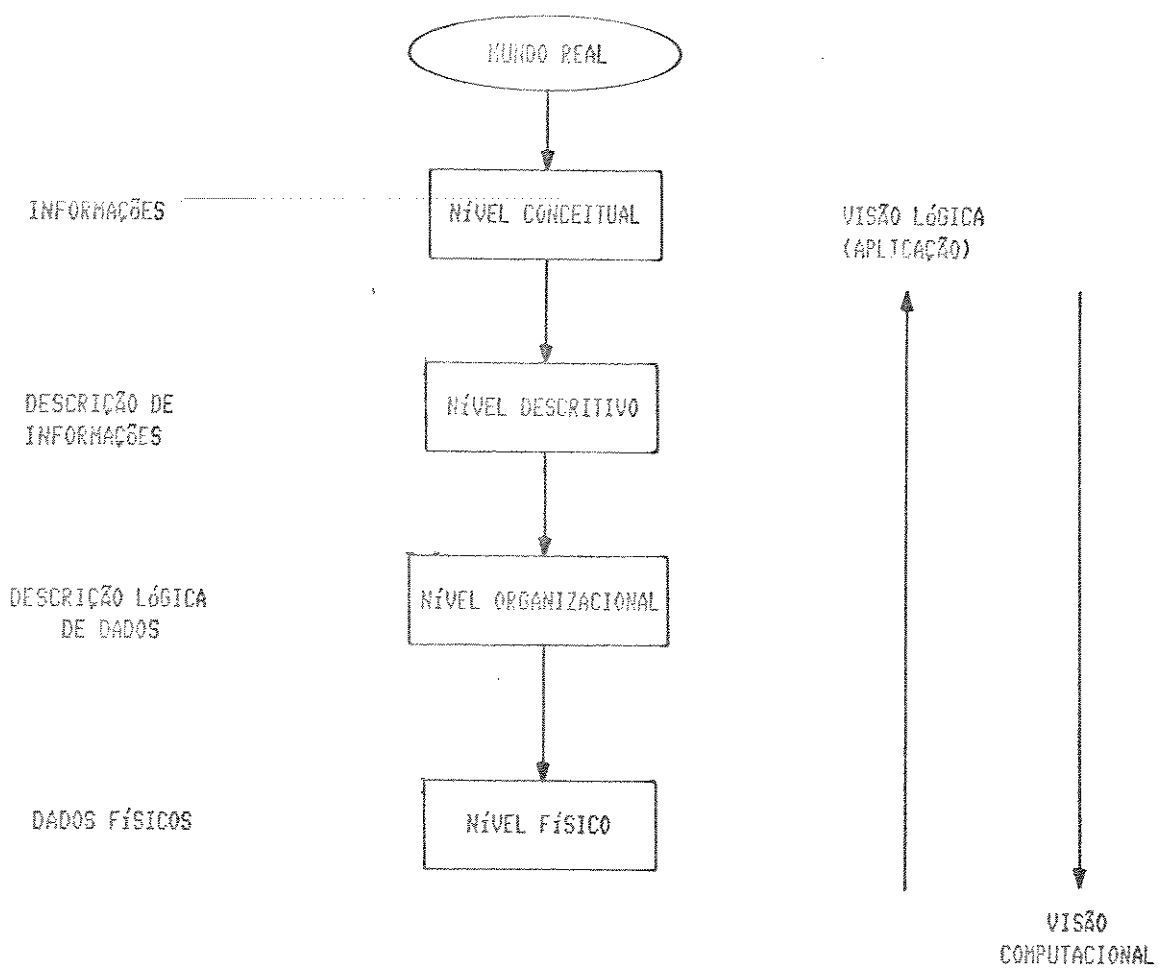


Figura 2.3. Níveis de Abstração de Dados

Os níveis Conceitual e Descritivo referem-se a uma percepção mais próxima do Mundo Real, enquanto os níveis

Organizacional e Físico estão mais próximos da concepção do Mundo Computacional. Através de uma sequência de ações conhecida como MAPEAMENTO, as informações modeladas na mente dos usuários (Nível Conceitual) serão transformadas em dados físicos armazenados nas memórias físicas dos computadores (Nível Físico).

2.4.1. Nível Conceitual

Neste nível o usuário lida com as informações do Mundo Real que dizem respeito a um determinado problema. Em outras palavras, apenas uma parcela do Mundo Real (aquela que interessa ao usuário) é modelada, utilizando-se uma forma de comunicação mais próxima e familiar ao usuário, preferivelmente sem (ou com um mínimo de) uso de formalismos matemáticos.

Assim, neste nível a descrição de um universo (ou suas partes) deve ser totalmente inteligível para as pessoas que interagem com ele, sem se exigir um conhecimento adicional ao que normalmente empregam nessa interação. Esta descrição deve ser a mais organizada possível e constitui um modelo da realidade, modelo este que será aqui denominado de modelo conceitual.

Existem diversas formas de comunicação para representar o modelo conceitual como, por exemplo, através de sentenças obedecendo as regras gramaticais e semânticas de cada língua, como no exemplo abaixo, que apresenta uma descrição textual simplificada da organização de uma empresa que projeta e fabrica peças de automóveis:

1. a empresa possui vários empregados;
2. cada empregado está vinculado a um departamento da empresa;
3. cada departamento é responsável por um projeto;
4. cada projeto utiliza (na fabricação do produto projetado) várias peças oferecidas por diversos fornecedores;
5. os empregados são caracterizados na empresa pelo nome, no. de inscrição no INAMPS e salário;
6. os projetos são caracterizados dentro da empresa pelo nome, número de projeto, data de início e tempo previsto de execução.

Com a crescente importância do computador como ferramenta de apoio às atividades humanas, tornaram-se necessários meios mais apropriados de comunicação entre os homens e os computadores. Idealmente, esta comunicação deveria ser em linguagem natural dos usuários (e.g., português, inglês, etc). Porém, no estágio atual do desenvolvimento do software, os computadores não adquiriram ainda a capacidade de lidar com vocabulários volumosos e estruturas gramaticais sofisticadas, havendo assim a necessidade de limitar a semântica da sua "linguagem" e estabelecer regras gramaticais mais simplificadas.

2.4.2. Nível Descritivo

Neste nível de abstração é desenvolvido um modelo da Informação denominado modelo descritivo. Já que o objetivo é chegar-

se, em um nível de abstração posterior, a um modelo computacional que possa ser fornecido e processado por um computador, a linguagem usada durante o estabelecimento do modelo descritivo deve ser estritamente formal.

Um modelo descritivo primário do modelo conceitual (problema) pode ser formalizado, por exemplo, através das seguintes primitivas: ENTIDADES (substantivos), RELACIONAMENTOS (verbos ou ações) e ATRIBUTOS (adjetivos e advérbios).

As ENTIDADES são os objetos e/ou eventos do modelo conceitual. É comum agrupar as entidades similares em conjuntos ou classes de entidades, denominadas TIPOS DE ENTIDADES. Cada tipo de entidade é qualificado ou caracterizado por um conjunto de ATRIBUTOS, que corresponde ao conjunto das características e propriedades dos objetos.

Em quase todas as situações reais, existem muitos tipos de entidades relevantes (de interesse aos usuários) que, quando associadas, compõem outras informações. A associação que se estabelece entre entidades é denominada RELACIONAMENTO. Essencialmente, classificam-se três tipos de relacionamentos - 1:1, 1:N e M:N. Considerando as associações entre dois tipos de entidades X e Y como correspondências

$$f_x: X \rightarrow Y$$

e

$$f_y: Y \rightarrow X,$$

os três tipos de relacionamentos podem ser definidos formalmente

como (/TSIG 77/) (Figura 2.4):

- * 1:1, quando ambas as correspondências f_x e f_y são funcionais (total ou parcialmente funcionais);
- * 1:N, quando somente uma das correspondências f_x ou f_y é funcional;
- * M:N, quando nenhuma das duas correspondências é funcional, isto é, cada 'x' pode ter muitos 'y' associados e vice-versa.

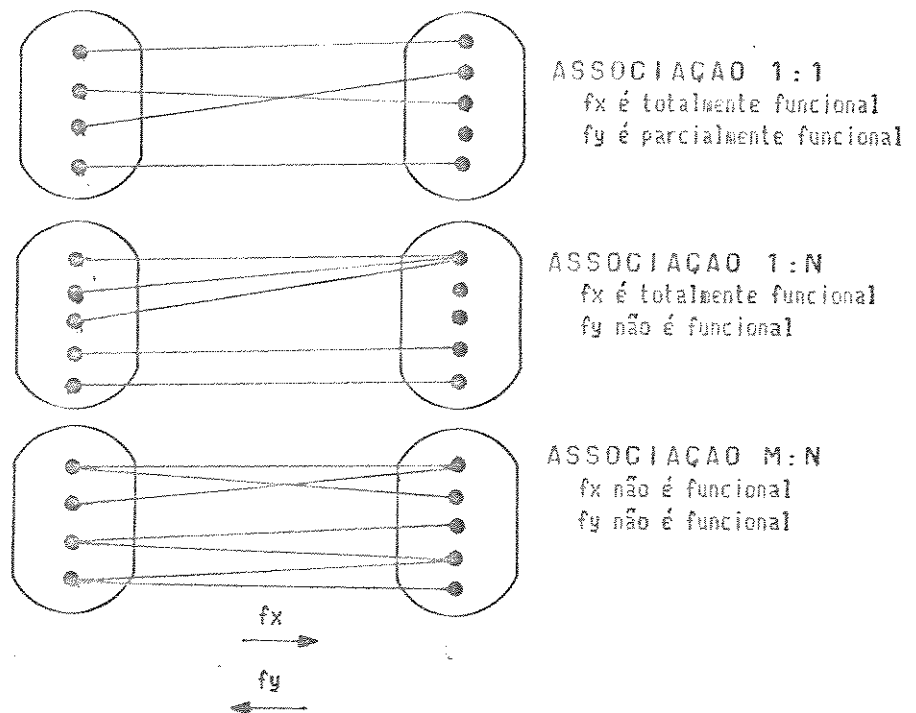


Figura 2.4. Tipos de relacionamentos

Este conceito de funcionalidade das associações torna explícito o tipo de relacionamento (1:1, 1:N, M:N) que existe entre os componentes identificados num contexto de aplicações.

Tendo como elemento inicial a formalização obtida

anteriormente, o usuário está apto a descrever as informações que lhe são pertinentes de uma maneira formal e precisa e, assim, transmitir estas informações ao responsável (ou responsáveis) pelo Sistema de Gerência de Banco de Dados (por exemplo, o ABD), de modo que este possa, a partir desta descrição, projetar a estrutura dos dados para a aplicação em estudo, utilizando as ferramentas oferecidas pelo Sistema de Gerência de Banco de Dados disponível.

Para isso, o ABD deve conhecer os tipos e a finalidade das informações que o Sistema de Banco de Dados armazenará. Como estes tipos de informação são específicos, dependentes das aplicações, eles são usualmente definidos pelo usuário especializado na aplicação. Com isso, surgiu um problema quanto à exatidão da comunicação, ao ABD, das informações que o usuário concebeu (modelo conceitual ou modelo descritivo primário). Como decorrência o ABD pode interpretar tais informações sob ângulos diferentes e decidir, em níveis posteriores de abstração, por uma organização ineficiente para as exigências dos usuários.

Como solução a esta possível ambiguidade de interpretação das informações sobre o Mundo Real, definidas no Nível Conceitual, são utilizados os MODELOS DE DADOS, através dos quais se especifica formalmente os requisitos de uma determinada aplicação e se define detalhadamente as primitivas identificadas no modelo descritivo. Em outras palavras, Modelos de Dados direcionam o usuário a descrever suas exigências numa forma concisa e precisa. Assim, os ABD's terão uma noção global e correta sobre a natureza das informações que devem constar na Base de Dados, para suportar os prováveis programas

que a utilizarão, e definir uma estrutura adequada aos recursos oferecidos pelo SGBD escolhido.

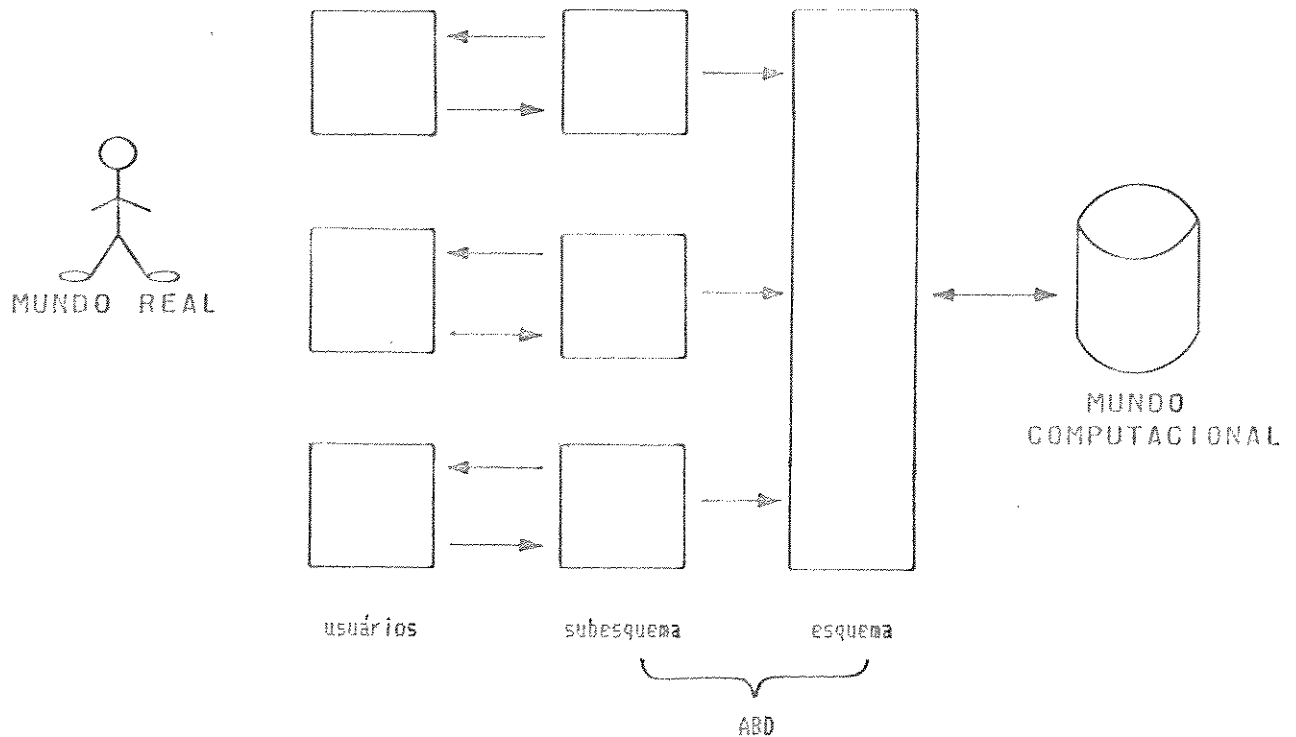


Figura 2.5. Interface entre os usuários (Mundo Real) e ABD (Mundo Computacional)

Assim, tem-se que o usuário utiliza um Modelo de Dados que lhe seja adequado para definir o modelo descritivo de seu problema. Por outro lado, o ABD também utiliza um Modelo de Dados suportado pelo Sistema de Gerência de Banco de Dados para que possa passar ao computador um modelo das informações de interesse. Dependendo da capacidade de abstração do SGBD, o mapeamento do modelo descritivo do usuário ao modelo submetido ao SGBD pode ser, ou não, imediato. Quando for imediato, os ABD's definem o "lay-out" da estrutura dos dados na Base de Dados utilizando diretamente os esquemas do Modelo de Dados fornecidos pelos usuários. Estes esquemas serão compilados

e usados pelo SGBD, tanto para codificar e decodificar corretamente os dados, como também para controle de integridade e consistência destes dados. Para o segundo caso, as informações devem ser transformadas e estruturadas em recursos aceitáveis pelo SGBD e a documentação da elaboração dos esquemas constitui uma importante fonte de referência para os usuários de como se deve manipular os dados armazenados (figura 2.5). Estes esquemas correspondem ao Nível Organizacional na hierarquia de Abstração de Dados.

2.4.3. Nível Organizacional

Este nível reflete a capacidade do SGBD de produzir automaticamente um mapeamento correto entre o Nível Físico (os dados organizados na Base de Dados) e as representações lógicas dos dados. Estas representações lógicas são utilizadas pelo ABD para organizar os dados em uma estrutura compatível com a oferecida pelo SGBD a partir das informações especificadas pelos usuários através dos Modelos de Dados correspondentes ao Nível Descritivo. Classificam-se os modelos do Nível Organizacional em duas categorias quanto ao seu grau de abstração:

- * dependente dos caminhos lógicos de acesso;
- * independente dos caminhos lógicos de acesso.

Os Modelos de Dados Independentes dos caminhos lógicos de acesso (por ex.: Modelo Relacional) não oferecem recursos explícitos para os usuários interferirem na organização física dos dados. Normalmente, um SGBD que os suporte é mais complexo, devendo conter algoritmos para mapear as descrições formais em estruturas físicas

mais convenientes aos programas de aplicação.

Qualquer que seja o grau de abstração, uma Base de Dados pode ser acessada por diferentes usuários com diferentes visões da Informação armazenada. Portanto, para facilitar a interação usuário/Sistema e proteger os dados de acessos inadequados, é interessante que se considere à parte as informações úteis a um determinado usuário das informações globais. Assim, sob este enfoque, destacam-se ainda dois subníveis de abstração (figura 2.6):

- * SUBESQUEMAS: referem-se às descrições formais dos dados utilizados pelos programas de aplicação. É a visão que cada usuário tem sobre a Base de Dados;
- * ESQUEMAS: referem-se às descrições formais do contexto global da Base de Dados. A partir dela, diferentes subesquemas podem ser derivados. É a visão que um ABD deve ter sobre a Base de Dados, para poder analisar e supervisionar o desempenho do Sistema como um todo.

Tanto os esquemas quanto os subesquemas constituem o denominado conjunto de MÁSCARAS utilizadas pelo SGBD para mapear dos dados físicos às abstrações correspondentes aos diferentes níveis/subníveis e vice-versa (figura 2.6).

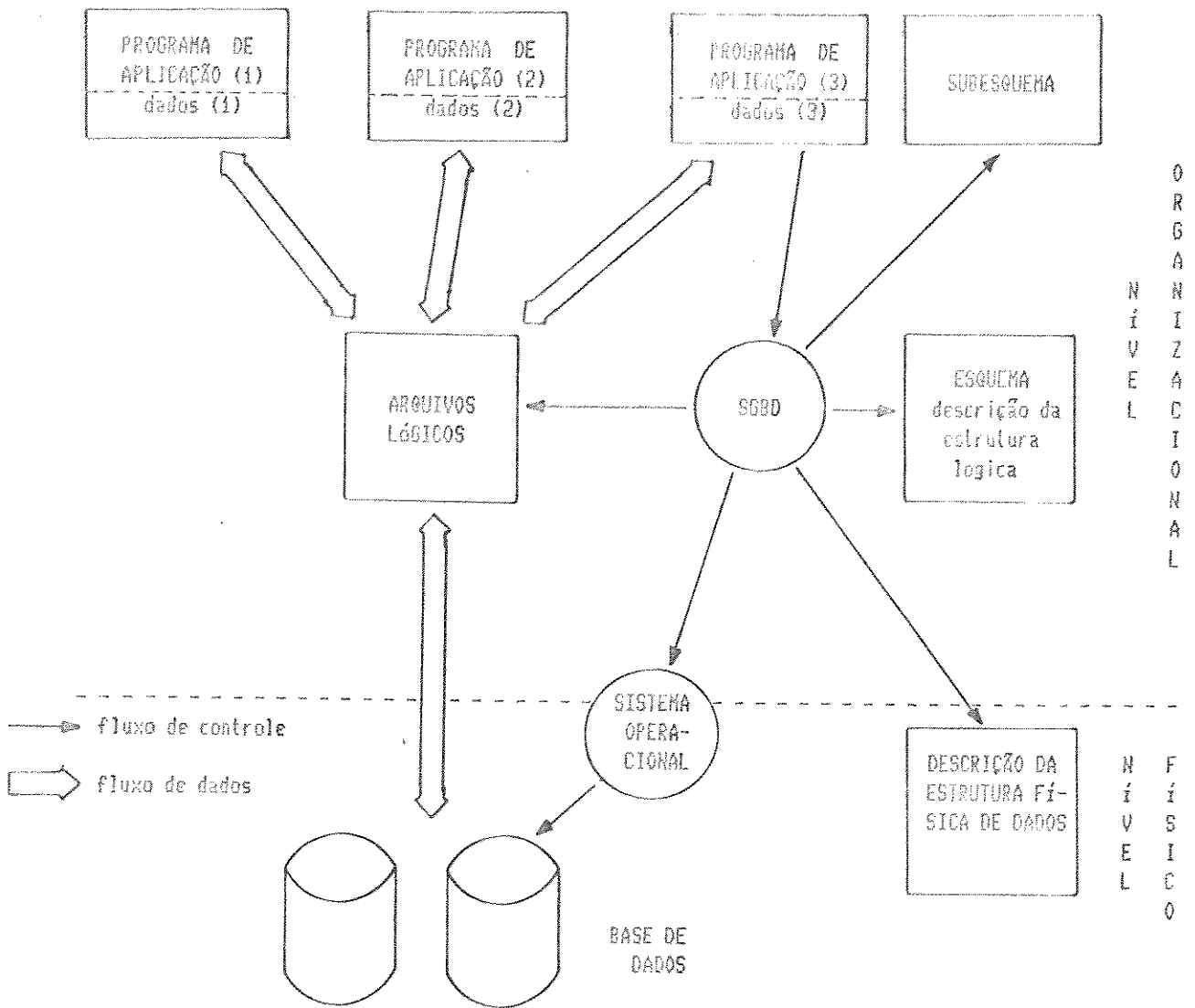


Figura 2.6. Distinção de dois subníveis de abstrações no Nível Organizacional (esquemas e subsquemas)

2.4.4. Nível Físico

Os critérios usados para definir a organização física dos dados são diferentes daqueles usados para a organização lógica. Na maioria dos sistemas, a implementação da estrutura física dos dados fica a cargo do Sistema Operacional sobre o qual o Sistema de Gerência de Banco de Dados está apoiado. Assim, somente serão

citadas as características físicas das Bases de Dados que devem ser analisadas pelo ABD para verificar o grau de compatibilidade entre o SGBD e as exigências das Aplicações.

É importante ressaltar aqui que muitos sistemas operacionais existentes não são realmente apropriados para serem utilizados especificamente como suporte a Sistema de Banco de Dados. Portanto, já existem estudos no sentido de desenvolver Sistemas Operacionais que propiciem suporte adequado a isso (/CARV 82/).

A organização física dos dados está intimamente ligada à eficiência operacional: tempo de operação (caminhos de acesso, localidade física dos dados, etc.) e custo (volume ocupado do espaço físico). Abaixo são relacionadas algumas características que devem ser verificadas na escolha de uma estrutura física dos dados (/MART 77/):

- * espaço de armazenamento;
- * redundância física de dados;
- * tempos de operação desejáveis (inserção, remoção, leitura, etc);
- * tempos de resposta desejáveis;
- * volatilidade dos dados armazenados: tempo de permanência dos dados no Sistema;
- * tipos de dados armazenados: tamanho dos registros, bloco, etc;
- * estrutura dos dados lógicos que ela suporta;
- * grau de confiabilidade e eficiência necessário;
- * custo.

Note-se que muitas destas características são incompatíveis entre si. Por exemplo, sistemas de alta densidade de armazenamento (pequeno espaço físico ocupado) requerem maiores tempos de busca (tempo de operação maior) (figura 2.7.).

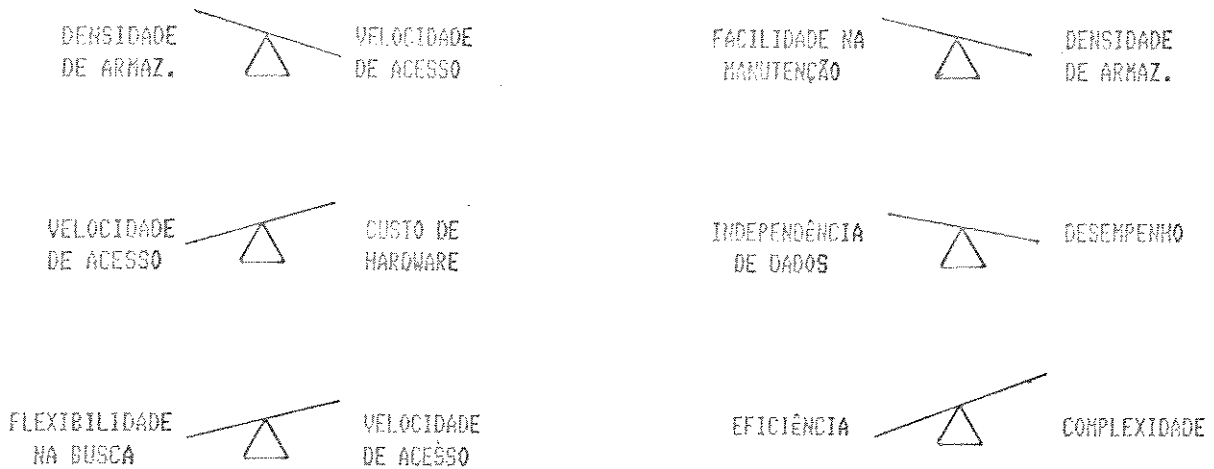


Figura 2.7. Características antagônicas de uma estrutura física

2.5. Modelos de Dados

Um MODELO DE DADOS é:

- * um conjunto de diretivas que padroniza a forma de representação da estrutura lógica dos dados contidos na Base de Dados (/TSIC 77/);
- * a visão que os usuários têm sobre a Base de Dados (/DATE 84/);
- * uma ferramenta para os usuários descreverem precisamente a sua percepção do Mundo Real

(/SMIT 80/).

A utilização de Modelos próximos à concepção do Mundo Real (programas de aplicação) é mais natural aos usuários não especializados em computação. A transformação desta concepção do Mundo Real em dados físicos armazenados na Base de Dados pode ser realizada automaticamente pelo SGBD ou através do Administrador de Banco de Dados (ABD), que mapeia manualmente as especificações formais dos usuários (Modelos do Nível Descritivo) em Modelos de Dados suportados pelo SGBD (/KAIN 80/).

O conjunto de formalizações que permite a estruturação e manipulação dos dados conforme a forma estabelecida pelo Modelo de Dados é denominado LINGUAGEM DE DADOS. Linguagens podem ser classificadas em três grupos (/TSIG 77/):

- * Linguagens de Definição de Dados (LDD);
- * Linguagens de Manipulação de Dados (LMD);
- * Linguagens de Definição de Armazenamento (LDA).

Uma LINGUAGEM DE DEFINIÇÃO DE DADOS descreve e define a estrutura dos dados relevantes aos programas de aplicação. Esta linguagem deve comportar algumas das seguintes capacidades:

- * especificar o modelo de dados, segundo o qual a Base de Dados é organizada pelo SGBD (se vários modelos estiverem disponíveis no Sistema);
- * atribuir identidade à Base de Dados e especificar as unidades lógicas que a compõem como, por exemplo, entidades, atributos, etc;

- * Identificar e descrever as associações que existem entre as unidades lógicas dos dados;
- * especificar os atributos dos tipos de entidades que são utilizados como chaves de identificação;
- * especificar as restrições de Integridade e/ou consistência;
- * especificar as regras de acesso; etc.

O grupo de operações para manusear os dados organizados na Base de Dados é exprimido através da LINGUAGEM DE MANIPULAÇÃO DE DADOS. A operação básica de manipulação é a SELEÇÃO DE DADOS (/TSIG 77/), que consiste em um conjunto de procedimentos para identificar, localizar e isolar os dados necessários para efetuar neles operações, tais como, inserção, remoção, modificação, consultas, etc. Distinguem-se dois tipos de critérios utilizados pelo SGBD para selecionar logicamente os dados:

- * endereçamento pelo conteúdo: quando a seleção é feita através dos valores dos atributos (ou qualificações) das entidades. Por exemplo, seleção de todas as entidades do tipo de entidade <FORNO>, que tem como valor de atributo <ESTADO DE OPERAÇÃO=NORMAL>;
- * endereçamento pela associação: quando a seleção se baseia nas associações (relacionamentos) estabelecidas entre os tipos de entidades. Por exemplo, obter os dependentes do empregado JOSÉ MARIO, onde <EMPREGADOS> e <DEPENDENTES> são classificados como tipos de entidades distintos.

Sob o ponto de vista de acesso físico dos dados, estes dois critérios de seleção se reduzem a um problema comum: a BUSCA ("searching" - /KNUT 73/). Existem diferentes técnicas para implementar um processo de BUSCA: listas invertidas, Quad-trees, B-trees, busca exaustiva, etc. As técnicas que são incluídas no SGBD são transparentes aos usuários comuns. Normalmente, cabe aos ADB's a responsabilidade de seleccionar os recursos de busca mais apropriados e estabelecer a correspondência entre os comandos da LMD e os recursos escolhidos. Portanto, os aspectos da linguagem de manipulação de dados referentes a um Modelo de Dados não indicam necessariamente os mecanismos de acesso físico/lógico.

É possível distinguir ainda em alguns modelos de dados uma terceira classe de comandos que descreve as estruturas de armazenamento físico, tais como: área de armazenamento, localidade dos registros lógicos, etc. Estes comandos são enquadrados na categoria de LINGUAGEM DE DEFINIÇÃO DE ARMAZENAMENTO (LDA), que é capaz de:

- * seleccionar o dispositivo de armazenamento (se existir mais de um disponível);
- * descrever o mapeamento entre a estrutura lógica e a estrutura física dos dados;
- * especificar os identificadores utilizados para acessar fisicamente as unidades lógicas dos dados;
- * especificar a ordenação física dos dados;
- * especificar o tipo de conversão dos formatos de dados. Por exemplo, binário -> inteiro.

A seguir, serão descritos sucintamente três modelos de dados que apresentam níveis de abstração diferentes: Modelo Rede, Modelo Relacional e Modelo Entidade-Relacionamento.

2.5.1. Modelos de Dados - Rede e Relacional

2.5.1.1. Modelo Rede

O MODELO REDE é um dos modelos de dados que mais se aproxima do Mundo Computacional, ou seja, da estrutura organizacional física dos dados. A proposta clássica do Modelo Rede originou-se do relatório do Grupo CODASYL do DBTG (/CODAS 71/), cujos conceitos e definições servirão de base de descrição do Modelo Rede que se segue.

No Modelo Rede, distinguem-se duas primitivas: TIPOS DE REGISTROS e TIPOS DE CONJUNTOS. No mapeamento do Nível Descritivo ao Nível Organizacional, normalmente os TIPOS DE REGISTROS são utilizados para representar os tipos de entidades, definidos pelos usuários, e os TIPOS DE CONJUNTOS para representar as associações entre os tipos de entidades.

Os TIPOS DE REGISTROS são caracterizados pelos ITENS DE DADOS e os valores (OCORRÊNCIA DOS ITENS DE DADOS) assumidos pelos itens definem unidades distintas, denominadas OCORRÊNCIAS DE REGISTRO, de um mesmo tipo de registro. Um grupo de valores de itens que define univocamente ocorrências de um tipo de registro é

designado CHAVE do tipo de registro.

Neste Modelo, a correspondência entre dois tipos de registros distintos é denominada TIPO DE CONJUNTO, onde um dos registros é denominado TIPO DE REGISTRO PROPRIETÁRIO ("owner") e o outro, TIPO DE REGISTRO MEMBRO ("member"). A associação definida no Modelo Rede representa a abstração da conexão física existente entre os dados na Base de Dados. Para obter os registros de seu interesse, os usuários devem seguir os caminhos estabelecidos entre os registros pelos elos (conexões) na Base de Dados. Este tipo de acesso é conhecido como NAVEGAÇÃO - processo em que se seguem explicitamente os caminhos de acesso para obter as informações desejadas.

Cada associação entre a ocorrência do tipo de registro proprietário e um conjunto de ocorrências do tipo de registro-membro, (este conjunto deve ter quantidade de elementos ≥ 0) é designada uma OCORRÊNCIA DO CONJUNTO, devendo satisfazer as seguintes regras de correspondência (/TAYL 76/):

1. dado um registro-proprietário, é possível acessar os registros-membros;
2. dado um registro-membro, é possível obter o seu registro-proprietário;
3. dado um registro-membro, é possível obter os outros registros-membros pertencentes à mesma ocorrência do tipo de conjunto;
4. cada registro-membro só pode estar associado a um registro-proprietário em um tipo de conjunto.

O Modelo Rede suporta os três tipos de relacionamentos: 1:1, 1:N, M:N. Para ser possível a modelagem de relacionamentos M:N (que viola a regra [4] acima) e para permitir a representação de relacionamentos recursivos, o Modelo CODASYL define um terceiro tipo de registro: o PSEUDO-REGISTRO ou TIPO DE REGISTRO RELACIONAMENTO (/FURT 79/, /TAYL 76/).

Como o Modelo Rede apresenta um grau de abstração que se enquadra dentro do Nível Organizacional, são incluídos explicitamente nas suas linguagens de dados comandos referentes aos pormenores dos acessos que foram estabelecidos entre os registros para obter as informações desejadas, tais como: FIND NEXT, GET NEXT, GET, GET...VIA..., MOVE...TO..., etc. Deste modo, a eficiência das operações de manipulação programadas, usando os recursos disponíveis na LMD, dependerá em grande parte do conhecimento dos usuários sobre os detalhes da estruturação lógica do Banco de Dados.

Com a LDD, o ABD define e declara todos os tipos de dados que irão compor a Base de Dados. A visão global do ABD de todas as descrições referentes aos dados é conhecida como ESQUEMA do Modelo, enquanto que a visão parcial, correspondente aos dados de interesse aos usuários, é chamada SUBESQUEMA.

2.5.1.2. Modelo Relacional

O MODELO RELACIONAL é um modelo baseado na teoria matemática de relações para representar as associações tanto entre

os atributos de uma mesma entidade como entre os de diferentes tipos de entidades. Por exemplo, (FABRICANTE, No. DA SÉRIE, CAPACIDADE, CARGA MÁXIMA) é uma associação dos atributos do tipo de entidade <MOTOR> e (ACIONADOR, TIPO DE CARGA, MOTOR) seria um exemplo da associação entre três tipos de entidades: MOTOR, CARGA e PESSOA.

Cada RELAÇÃO é um conjunto de tuplas definidas em um conjunto de 'n' domínios C_1, C_2, \dots, C_n , não necessariamente distintos. Os valores que definem cada elemento de uma tupla são chamados COMPONENTES da tupla. Um conjunto de componentes referentes a uma tupla define uma OCORRÊNCIA. Graficamente, uma relação n-ária R pode ser representada como uma tabela, onde, entre outras coisas (/CDDD 70/), cada linha representa uma tupla da relação R (ocorrência) e todas as linhas são distintas, ou seja, duas tuplas diferem em pelo menos uma componente.

A coluna ou conjunto de colunas (domínios), cujos valores identificam univocamente cada elemento (tupla) em uma relação é denominada CHAVE-CANDIDATA. É possível que uma relação tenha duas ou mais chaves-candidatas. Neste caso, seleciona-se arbitrariamente uma destas chaves como a CHAVE PRIMÁRIA, ou simplesmente CHAVE, da relação (/CDDD 72/).

No Modelo Relacional, não há declaração explícita das associações entre diferentes relações (lembrando-se que nele só existe basicamente uma primitiva: relação). A referência das tuplas de uma relação às tuplas de uma outra relação é feita através da REFERÊNCIA CRUZADA, usando o conceito usualmente denominado de CHAVE

IMPRÓPRIA ("foreign key").

O Modelo Relacional é um modelo que apresenta um grau de abstração de dados maior que o Modelo Rede, pois nele é suposto que os ABD's não precisam ter conhecimento dos detalhes de endereçamento lógico dos dados para acessá-los. Por exemplo, os procedimentos de navegação citados para o modelo rede ficam transparentes aos seus usuários. Assim, a sua linguagem de dados é menos procedural que a do Modelo Rede e muitos autores preferem classificá-la como linguagem de consulta ("query language") (/ULLM 80/, /FRY 76/). Destacam-se três tipos clássicos de representação desta linguagem:

- * ÁLGEBRA RELACIONAL (/CODD 70/, ULLM 80/, /DATE 84/);
- * CÁLCULO RELACIONAL SOBRE TUPLAS (ULLM 80/, /DATE 84/);
- * CÁLCULO RELACIONAL SOBRE DOMÍNIOS (ULLM 80/, /DATE 84/).

A descrição do universo de dados utilizando a LDD é mais simples no Modelo Relacional que no Modelo Rede, pois é suposto que o SGBD se encarrega de todas as operações referentes à organização física e lógica dos dados, de modo que as funções dos ABD's se restringem à declaração dos tipos de dados que serão armazenados e a definição de alguns detalhes específicos de aplicação.

O projeto da estruturação dos dados, segundo o modelo relacional, baseia-se na TEORIA DE NORMALIZAÇÃO, que consiste em um conjunto de diretivas para escolher dentre as possíveis alternativas

dos esquemas do Banco de Dados, aquela que apresenta propriedades mais favoráveis: menor redundância, menor probabilidade de ocorrência de anomalias de atualização, inserção, remoção, etc (/ULLM 80/). A teoria se fundamenta na redução do universo de relações ao conjunto de relações normalizadas. As formas mais conhecidas de normalização são: primeira, segunda e terceira formas normais (/CODD 72/), forma normal BOYCE-CODD (/CODD 74/), quarta forma normal (/FAGIN 77/) e quinta forma normal (/FAGIN 79/).

2.5.2. Modelo Entidade-Relacionamento

O Modelo Entidade-Relacionamento (ou simplesmente MER) enquadra-se no conjunto de Modelos de Dados que apresentam grau de abstração correspondente ao nível Descritivo. Ele dá maior ênfase à definição da semântica de Informações, direcionando os usuários a definir com precisão e numa forma natural, em relação à sua concepção usual, os seus requisitos. Neste capítulo serão adotados os conceitos MER desenvolvidos em /CHEN 76/, /CHEN 77/ e /NG 81/, conceitos estes que serão expostos a seguir.

2.5.2.1. Conceitos Gerais

No MER distinguem-se dois grupos de primitivas: ENTIDADES e RELACIONAMENTOS. A ENTIDADE é um objeto que tem existência própria. Um grupo de entidades similares ou seja, que apresentem propriedades comuns, é denominado CONJUNTO DE ENTIDADE ou TIPO DE ENTIDADE. O RELACIONAMENTO define a associação entre as entidades por meio das FUNÇÕES ou PAPÉIS ("role") desempenhadas por elas nesta associação.

O CONJUNTO (ou TIPO) DE RELACIONAMENTO define todas as associações estabelecidas entre 'n' ($n \geq 1$) entidades, cada qual pertencente a um conjunto de entidades, conjuntos este que não são necessariamente distintos.

Tanto entidades quanto relacionamentos podem ser caracterizados em termos de um conjunto de valores. Estes valores são classificados em diferentes DOMÍNIOS DE VALORES. A correspondência entre entidades/re relacionamentos e os valores é feita através de ATRIBUTOS. Formalmente, um atributo é definido como uma aplicação que mapeia de um tipo de entidade ou relacionamento para um domínio de valores (ou produto Cartesiano destes). Se esta aplicação é funcional o atributo é monovalorado, e se a aplicação é não-funcional o atributo é multivalorado (/NG 81/).

É importante notar que no Modelo Entidade-Relacionamento, um tipo de relacionamento é mais que uma conexão lógica. Ele deve ser considerado como um elemento concreto com existência própria, da mesma forma que as entidades, isto é, sem necessariamente estar estabelecendo associação entre tipos de entidades. Da mesma forma que entidades, tipos de relacionamentos podem ter atributos, conceito este que é importante no entendimento da semântica de dados e na determinação das dependências funcionais entre dados (/CHEN 76/).

Em /CHEN 76/ é utilizado, a nível organizacional (portanto, fora do escopo do MER) o conceito de chave de entidade, que consiste de um grupo de atributos tal que o mapeamento de um tipo de entidade

sobre o grupo correspondente de domínio de valores é um-para-um. Em outras palavras, a chave é um conjunto de atributos, cujos valores nos domínios de valores correspondentes identificam unicamente a entidade. De forma análoga, define-se chave de relacionamento como sendo o conjunto de todas as chaves das entidades envolvidas no relacionamento. Deve-se notar que /CHEN 76/ situa o conceito de chave a nível de representação dos dados (nível organizacional) e, portanto, tal conceito não faz parte do escopo do MER, que é um modelo de dados que se situa no Nível de Abstração Descritivo.

Um conceito importante introduzido neste Modelo é a definição do relacionamento sobre um conjunto de funções, ao invés da sua definição sobre os domínios de valores, como ocorre no Modelo Relacional. Dentre as vantagens desta conceituação, destacam-se:

- * as funções desempenhadas pelas entidades num relacionamento são únicas, isto é, não há possibilidade da ocorrência de ambiguidade da interpretação de uma informação contida numa tupla;
- * um conjunto de relacionamentos pode estabelecer associações idênticas entre conjuntos de entidades distintos, desde que eles desempenhem as mesmas funções, o que se aproxima mais da concepção real, pois certas funções podem ser desempenhadas pelos diferentes conjuntos de entidades e vice-versa.

Uma outra noção que favorece o Modelo Entidade-Relacionamento é a classificação das entidades e relacionamentos em FRACO e REGULAR e, associado com esta classificação, o conceito de

DEPENDÊNCIA DE EXISTÊNCIA e DEPENDÊNCIA DE IDENTIFICAÇÃO (/CHEN 77/). Quando a existência de uma entidade depende da existência de uma outra (ou outras), diz-se que ocorre uma dependência de existência, sendo que a entidade dependente é denominada tipo de entidade fraca. Por outro lado, se uma entidade não pode ser univocamente identificada por seus próprios atributos e deve ser identificada pelos seus relacionamentos com outras entidades, diz-se que há uma dependência de identificação, notando-se que o tipo de entidade dependente é também um tipo de entidade fraca. Se uma entidade não tem sua existência dependente de outra e/ou pode ser identificada por seus próprios atributos, então tem-se um tipo de entidade regular.

No que diz respeito à classificação de relacionamentos em fracos e regulares, diz-se que um relacionamento é regular se todas as entidades envolvidas são identificadas por seus próprios atributos ou se este relacionamento não estabelece quaisquer uma das dependências citadas acima. Por outro lado, se alguma das entidades em um relacionamento é identificada por outros relacionamentos, diz-se que o relacionamento é fraco. Assim, em um relacionamento regular, todas as entidades envolvidas são regulares. Da mesma forma, um relacionamento é fraco se alguma das entidades envolvidas for uma entidade fraca, notando-se assim que o relacionamento que estabelece uma dependência de existência ou de identificação é também um relacionamento fraco. A vantagem destas definições é a facilidade que advém ao controle de consistência e integridade dos dados.

O MER não define uma LMD específica, embora /CHEN 76/ estabeleça alguns princípios e diretrizes que devem reger a manipulação de dados e a definição de restrições de integridade. No que se refere à LDD, /CHEN 76/ define uma técnica diagramática para a representação das primitivas MER, que é denominada Diagrama Entidade-Relacionamento (DER), ilustrado na figura 2.8.

Em um DER, o tipo de entidade regular (entidade CARRO na figura 2.8) é representado por um retângulo, enquanto que um relacionamento (relacionamentos POSSUI, FABRICADO, PRODUZ na figura 2.8) é representado por um losango. O tipo de entidade fraca (entidade RODA) é representado por um retângulo com contorno duplo, sendo que a seta associada à entidade fraca indica a direção da dependência de existência (entidade RODA é dependente em relação à entidade CARRO).

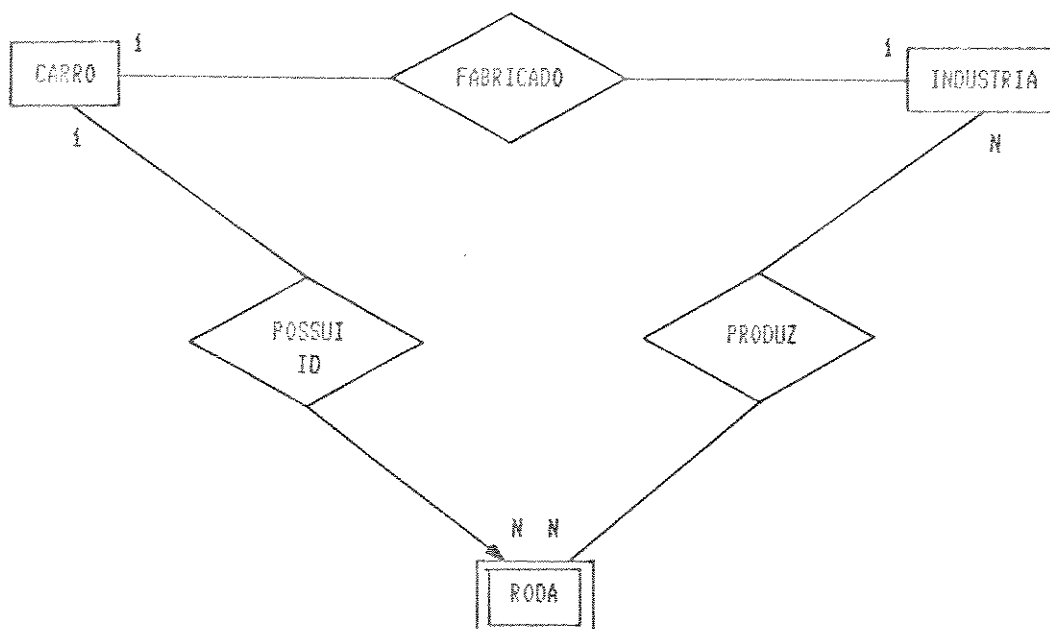


Figura 2.8. Primitivas MER - Diagrama E-R

A letra E dentro do losango indica que há apenas dependência de existência. A identificação ID em um relacionamento (relacionamento POSSUI) indica a dependência de identificação. Em ambos os casos, os relacionamentos são fracos. Observe-se que o relacionamento PRODUZ é também um relacionamento fraco (em relação ao relacionamento POSSUI), pois envolve entidades fracas (tipo de entidade RODA).

2.5.2.2. Estudos Adicionais sobre o MER

Desde o aparecimento do MER, têm surgido trabalhos que se dedicam ao estudo dos conceitos de Entidade-Relacionamento. Uma boa referência bibliográfica pode ser encontrada em /CHEN 80/, /CHEN 81/ e /DAVIS 83/, onde são mostrados e referenciados diversos trabalhos que procuram mostrar a utilidade dos conceitos do MER em áreas diversas, como projeto e análise de sistemas, modelagem e análise da informação e engenharia de software.

Este grande interesse que a abordagem Entidade-Relacionamento despertou desde a sua concepção é devido a uma série de fatores (/CHEN 83a/). Um dos fatores foi a necessidade de unificar os modelos de dados existentes. Como o MER é um modelo de dados que se aproxima bastante do nível conceitual de abstração, ele possui certas características interessantes que não existem em modelos de mais baixo nível, como os modelos relacional, rede e hierárquico.

Um outro fator de importância foi a necessidade de desenvolver uma metodologia de projeto de banco de dados que fosse

independente dos SGBD's comerciais existentes. A abordagem de entidades e relacionamentos surgiu como uma boa forma de se modelar o mundo real. Uma outra necessidade que surgiu foi a de se converter dados de um SGBD para outro SGBD, necessidade esta que se tornou mais premente com o surgimento de banco de dados distribuídos, onde muito provavelmente são usados diferentes SGBD's nas diferentes localidades da rede distribuída. O problema da conversão de dados surgido motivou então a pesquisa sobre uma forma padrão para a modelagem de dados, o que culminou com a definição do MER.

A partir do surgimento do MER, muitas direções de pesquisa foram aparecendo, direcionadas para dois tipos de problemas: convencionais e não-convencionais. Dentro dos problemas convencionais, as pesquisas têm se direcionado para:

- * uma formalização mais profunda do MER;
- * estudos e propostas de modelos de dados alternativos que utilizem os conceitos de "entidade" e "relacionamento";
- * desenvolvimento de SGBD's que suportem o MER (direta ou indiretamente);
- * extensões do MER visando um maior potencial para representação da semântica da informação;
- * propostas de linguagens baseadas em entidades e relacionamentos;
- * problemas relativos a banco de dados distribuídos.

As pesquisas sobre aplicações não-convencionais utilizando a abordagem Entidade-Relacionamento têm se direcionado para:

- * modelagem de conceitos temporais e históricos;
- * modelagem de aplicações não tradicionais como PAC/FAC, estruturas VLSI, sistemas de conhecimento, etc;
- * modelagem do comportamento dinâmico de entidades e relacionamentos.

Em vista deste reconhecimento geral da adequação da abordagem ER na descrição de problemas convencionais e não-convencionais, optou-se neste trabalho pela adoção desta mesma abordagem no estudo de aplicações PAC. Tal estudo se desenvolverá nos próximos capítulos, com a descrição dos aspectos envolvidos em aplicações PAC (capítulo 3), para em seguida utilizar os conceitos de Entidade-Relacionamento para a modelagem e descrição destes aspectos (capítulo 4).

Deve ser observado que na abordagem adotada na maioria dos trabalhos pesquisados, o MER é utilizado apenas para especificação e projeto de aplicações (e.g., Banco de Dados). Uma vez feito o projeto, faz-se um mapeamento para um modelo de dados que seja implementável em um computador, via de regra um modelo de nível organizacional (e.g., Modelo Relacional, Modelo Rede, etc.).

O que é considerado neste trabalho, contudo, é que o MER (ou extensões deste) pode ser ambientado em um computador, mesmo sendo um modelo de dados de nível descritivo. Uma vez assumida esta posição, o MER pode ser suportado por um Sistema de Gerência de Banco de Dados, que se encarrega de fazer o mapeamento de um esquema

MER para as estruturas organizacionais utilizadas pelo SGBD. Não há, portanto, a necessidade do mapeamento intermediário de um esquema MER para um esquema definido através de um outro modelo de dados, mais próximo do mundo organizacional.

3.1. Introdução

Atualmente, tanto em países desenvolvidos como em desenvolvimento, a economia tem se defrontado com problemas cada vez mais complexos no que se refere às exigências do mercado em termos de qualidade e custo dos produtos. Torna-se necessário, portanto, que produtos sejam fabricados em grandes quantidades e a baixo custo. A solução que se apresenta, no contexto atual, é a automatização do processo de produção (desde a fase de projeto até o produto acabado) tanto quanto possível e quando for possível.

Em vista da evolução dos sistemas computacionais, estes passaram a desempenhar um papel importante na evolução da automatização de processos, deixando de exercer apenas o papel de máquinas de calcular mais potentes. Entre as atividades automatizadas está o auxílio ao projeto e fabricação de produtos. Aqui, produtos, em um sentido amplo, são elementos de qualquer sistema (sistemas de transporte, centros hospitalares, processos de planejamento, etc.). Esta integração entre ciência de computadores e engenharia deu origem a novos ramos de atividade: PAC (Projeto Auxiliado por Computador), FAC (Fabricação Auxiliada por Computador), CAT ("Computer Aided Testing" - Teste Auxiliado por Computador), CAE ("Computer Aided Engineering" - Engenharia Auxiliada por Computador), etc.

Numa definição mais formal, entende-se por PAC como o uso do conhecimento em hardware e software de computadores, em análise de sistemas e em métodos de engenharia, com o propósito de especificação, projeto e implementação de produtos que são necessários à sociedade.

Tal sistema de computador, ou sistema PAC, possui os seguintes componentes funcionais: Banco de Dados, Base de Métodos e Subsistemas que executam as funções elementares do Sistema PAC: diálogo, E/S de dados, gráficos e programas de aplicação (figura 3.1).

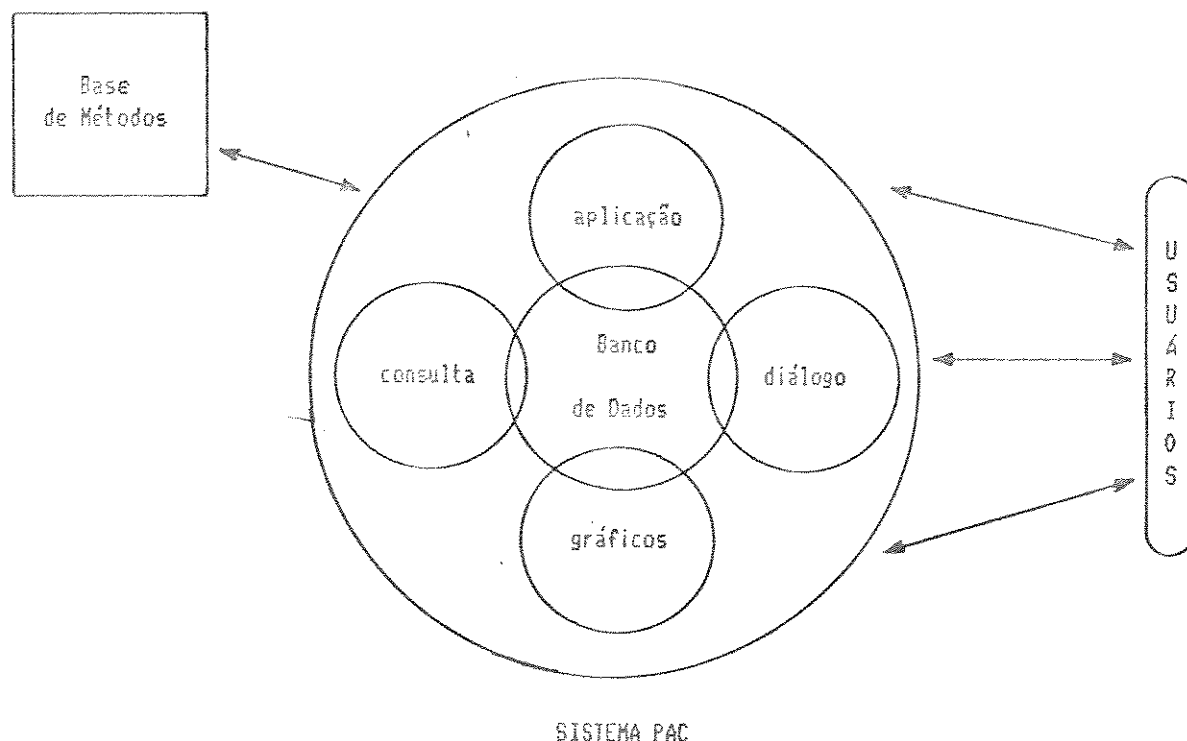


Figura 3.1. Componentes Funcionais de um Sistema PAC

A Base de Métodos consiste de uma biblioteca de programas com os módulos usados pelas funções elementares do sistema. Fazem

parte de um Subsistema de Comunicação os módulos para diálogo (comandos endereçados ao sistema PAC e mensagens devolvidas ao usuário), para entrada/saída de dados e para processamento de informações gráficas.

Diversas condições técnicas, econômicas, normativas e humanas afetam o desenvolvimento, distribuição e operação de sistemas PAC. Antes de empregar tais sistemas, os objetivos esperados em cada campo devem ser estabelecidos explícita e detalhadamente. Cada decisão deve ser tomada com respeito a estes objetivos para em seguida ser planejado um esquema cuidadoso de implementação do sistema com base nestes objetivos. Este capítulo procura dar uma descrição geral de arquitetura de sistemas PAC para em seguida estabelecer as principais características de aplicações PAC no que diz respeito à estruturação de dados de projeto.

3.2. Arquitetura de Sistemas PAC

A arquitetura de um sistema PAC depende de vários fatores, como:

- * a tarefa a ser executada pelo sistema;
- * os recursos computacionais disponíveis (hardware e software);
- * procedimentos pré-estabelecidos dentro de uma empresa ou em uma escala mais ampla (nacional e/ou internacional).

Sob o ponto de vista de hardware, existem dois tipos

principais de arquitetura :

- * sistemas 'stand-alone' ou independentes;
- * sistemas terminais (ligados a um computador hospedeiro), que podem ser classificados em: sistemas locais de Entrada e Saída (E/S), sistemas terminais inteligentes e sistemas satélites inteligentes.

Os sistemas locais de E/S possuem localmente apenas os dispositivos de E/S e uma interface para o computador hospedeiro. Os sistemas terminais inteligentes possuem, além dos dispositivos de E/S, seus controladores. Possuem também facilidades de processamento para executar a temporização de E/S e um processador para executar as subrotinas que acionam os dispositivos ("drivers"), a maioria das subrotinas do sistema e uma pequena parte do programa de aplicação.

Os sistemas satélites inteligentes podem ser vistos como sistemas de terminais inteligentes muito bem equipados. Mantêm localmente todas as rotinas do sistema e aproximadamente todo o programa de aplicação. O acesso ao computador hospedeiro é realizado a fim de aproveitar suas facilidades em termos de hardware e para acesso a dados e software disponíveis.

Pode-se observar que os sistemas PAC distribuídos oferecem um certo equilíbrio entre o uso de um sistema PAC dedicado e o uso de um sistema PAC central compartilhado no tempo. As vantagens oferecidas por um sistema distribuído (resposta rápida para pequenos problemas, apoio de um computador mais potente para a solução de

problemas complexos) são contrapostas, no entanto, pelo fato de que este tipo de sistema é muito mais complexo e de que o conceito de transportabilidade envolvido ainda está em desenvolvimento.

Sob o ponto de vista da flexibilidade do sistema, pode-se classificar os sistemas PAC em:

- * sistemas 'turn-key' são sistemas concebidos com um conjunto pré-determinado de funções que não podem ser adaptados livremente a um ambiente específico. Tais sistemas geralmente operam somente em um hardware definido.
- * sistemas livremente programáveis: são sistemas que contam com um conjunto básico de funções que podem eventualmente serem adaptadas ao ambiente no qual o sistema será utilizado. Esta adaptação ao ambiente pode ser feita via agregação de novas funções ao sistema e/ou através da restrição de parâmetros das funções básicas. Estes sistemas apresentam um elevado grau de transportabilidade, isto é, eles são facilmente adaptáveis a diferentes tipos de computadores (hardware).

3.2.1. Componentes de um Sistema PAC

Pode-se enfocar o problema de determinar os diversos componentes de um sistema PAC, sob três aspectos: funcional, hardware e software.

3.2.1.1. Componentes Funcionais

Este aspecto é esquematizado na figura 3.2, onde um sistema PAC é representado em função das atividades desenvolvidas durante um processo de projeto.

O grau de sofisticação de cada um dos componentes depende da aplicação em questão e a interface entre os mesmos é uma característica importante nas arquiteturas de sistemas PAC.

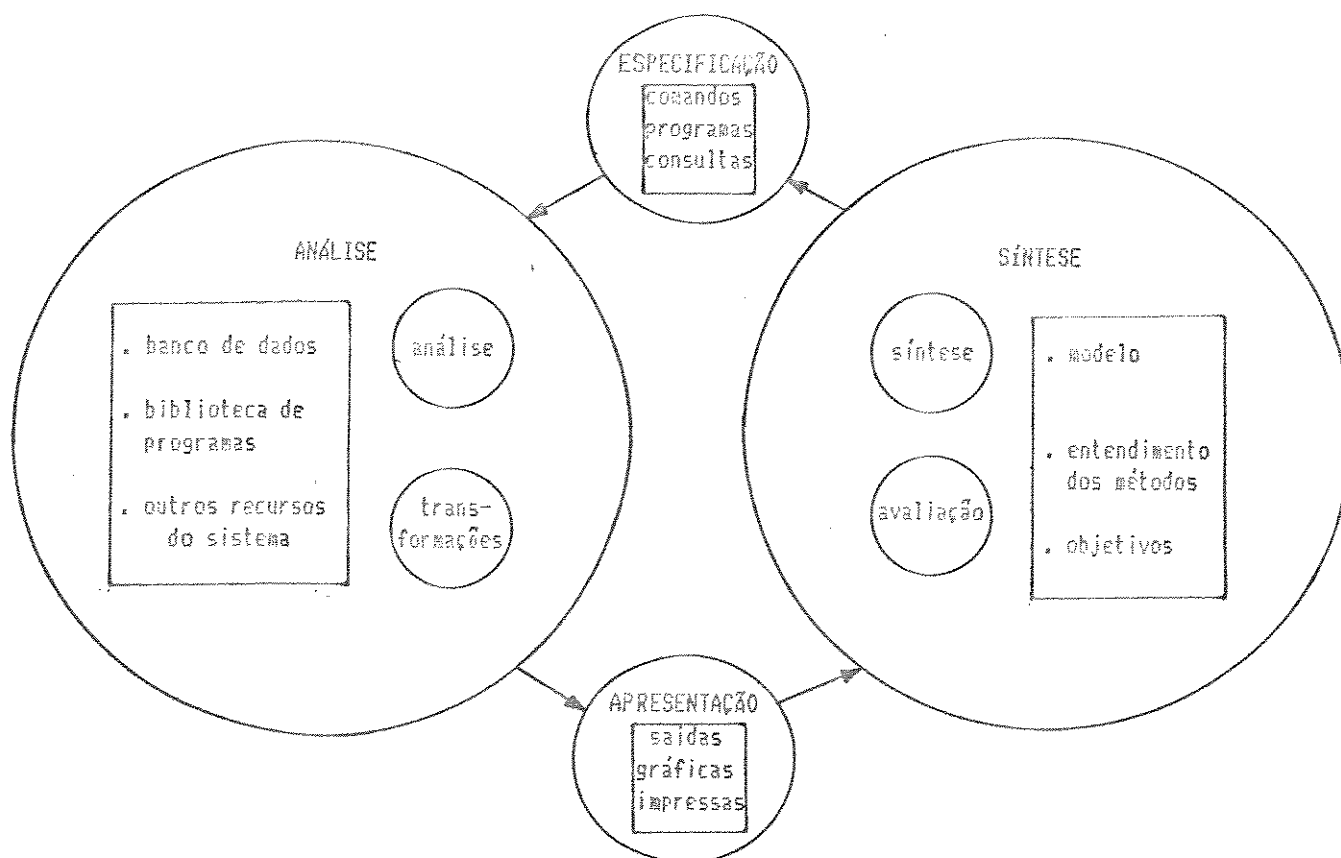


Figura 3.2. Componentes funcionais de um sistema PAC

As especificações do projeto são introduzidas no sistema PAC pelo usuário através de comandos, consultas ou programas

(ESPECIFICAÇÃO). Esses dados são então analisados e transformados adequadamente utilizando a base de métodos e outros recursos do sistema (ANÁLISE). Os resultados são apresentados aos operadores que os avaliam e comparam com os objetivos a serem atingidos (APRESENTAÇÃO e SÍNTESE). Usualmente estes resultados são apresentados sob a forma gráfica para facilitar a comunicação homem-máquina.

A Biblioteca de Programas contém os módulos das funções elementares do sistema (diálogo, E/S de dados, pacote gráfico) e os módulos que representam os algoritmos utilizados pelos programas de aplicação (figuras 3.1 e 3.2).

A eficiência do sistema PAC e a sua aceitação por parte dos usuários estão intimamente ligadas à interface de comunicação homem-máquina que inclui o módulo de diálogo (que se constitui basicamente de comandos enviados ao sistema e de mensagens recebidas pelo usuário), módulo para entrada e saída (E/S) de dados e módulo para processamento de informações gráficas.

O módulo de E/S executa as funções de entrada e saída de dados realizando testes para garantir a integridade e consistência dos dados. A entrada e a saída de informações gráficas, além do diálogo gráfico, são executados no módulo de processamento de informações gráficas.

3.2.1.2. Componentes de Hardware

Os componentes do sistema PAC, em termos do hardware, variam conforme a arquitetura adotada para o sistema. Além do sistema computacional convencional, o sistema poderá possuir vários tipos de memória secundária (dispositivos de armazenamento), como unidades de disco e de fita magnética, e dispositivos de entrada e de saída que permitem a comunicação homem-máquina.

Como dispositivos de entrada, pode-se ter um ou mais de um dos seguintes tipos: teclado, "light-pen", mesa digitalizadora e dispositivos de posicionamento como "mouse" e "Joystick". Quanto aos dispositivos de saída, existem duas classes principais: dispositivos não interativos e dispositivos gráficos interativos.

Os primeiros incluem os diversos tipos de plotadores e impressoras, como: plotadores eletrostáticos, plotadores a laser, impressoras do tipo 'hard-copy', etc.

Os dispositivos de saída para interação gráfica podem ser do tipo armazenador ("storage"), regenerador ("refresh") ou composto, conforme haja ou não necessidade de regenerar a imagem periodicamente. De acordo com a forma através da qual a imagem é gerada, estes dispositivos (monitores ou "displays") podem ser ainda do tipo raster ou vetorial. Como exemplo destes dispositivos temos o "storage tube", o painel de plasma, o "vector refresh display" e o "raster scan display" (/NEWM 83/).

3.2.1.3. Componentes de Software

Finalmente, é indispensável falar-se sobre o software de um sistema PAC. Ele é que provê os meios adequados para ativar e controlar as várias partes que compõem o hardware do sistema.

É possível dividi-lo em módulos elementares relacionados com as atividades funcionais desempenhadas por eles. Dessa forma, tem-se:

- * pacote para tratamento de diálogo;
- * pacote para gerência de dados;
- * pacote gráfico;
- * base de métodos, que consiste de um pacote de software de uso comum aos programas de aplicação e aos pacotes anteriores;
- * pacote de software específico para tratamento do problema de aplicação (algoritmos de otimização, de análise estrutural, etc).

Um ponto muito importante durante o projeto de um sistema PAC é a implementação do módulo de diálogo, cuja função principal consiste em interpretar os comandos enviados pelo usuário e iniciar a execução dos módulos de aplicação referentes àqueles comandos. Atualmente, muitos esforços estão sendo concentrados no sentido de desenvolver módulos de diálogo que tornem o uso do sistema PAC agradável aos usuários.

Da mesma forma que é possível definir um conjunto de

funções gráficas independente da aplicação em questão, é também possível estabelecer um núcleo para o diálogo. O projeto é deixado a cargo do projetista do sistema PAC, e a implementação de funções específicas da aplicação recai sobre o núcleo do sistema de diálogo. O módulo de diálogo apresentará, dessa forma, todas as funções necessárias a um determinada aplicação, obedecendo às regras do núcleo de diálogo, sua estrutura, interfaces e conceitos (/SILVA 85/, /HARTELT 86/).

Quanto aos pacotes gráficos, pode-se observar que existe uma grande variedade dos mesmos. A escolha de um em particular irá depender das necessidades de cada aplicação. O software gráfico é constituído de diversas camadas superpostas, com as seguintes funções:

- * CAMADAS INTERNAS (mais próximas do hardware, transparentes ao operador):
 - rotinas de comunicação do computador com o dispositivo gráfico;
 - rotinas gráficas elementares que se encarregam dos traçados gráficos simples: pontos, retas, arcos de círculo, caracteres, etc;
 - rotinas para criação de figuras elementares: polígonos, círculos, flechas, etc;
 - rotinas para gravação em disco e recuperação em sistemas de bancos de dados gráficos.

- * CAMADAS EXTERNAS (mais próximas do usuário):
 - rotinas para criação/manutenção de desenhos e

objetos:

- rotinas para visualização de desenhos: zoom, perspectivas de objetos tridimensionais, cortes, eliminação de linhas escondidas, etc;
- comandos especiais de uma aplicação particular.

As rotinas gráficas sobrecarregam a CPU dos computadores e por isso existe uma tendência na construção de dispositivos gráficos inteligentes, com o uso de microprocessadores locais que tendem a absorver, além das funções das camadas internas, uma parte das funções das camadas mais externas do software. Esta diversificação da capacidade dos dispositivos gráficos disponíveis no mercado tem dificultado a tarefa de desenvolvimento de um software gráfico genérico. A finalidade da construção deste tipo de software é proporcionar principalmente maior transportabilidade.

Vários trabalhos foram e estão sendo desenvolvidos no sentido de obter-se uma padronização no tratamento das informações gráficas sob o ponto de vista dos programas de aplicação, visando tornar tal tratamento independente dos dispositivos. Dentre os padrões gráficos propostos, tem-se: GKS (Graphical Kernel System - /ISO 82/ - já aceito como padrão internacional pela ISO) e PHIGS - (Programmers' Hierarchical Interactive Graphics Standard - /ISO 85/). É bom ressaltar aqui que nenhuma destas propostas de padronização definem com clareza a forma de tratamento da adaptação de um pacote gráfico aos diferentes dispositivos gráficos, exigindo a implementação de "drivers". Com o objetivo de suprimir esta deficiência, foi proposto o CGI (Computer Graphics Interface - /ISO

86/), que é um padrão de Interface com dispositivos gráficos que visa a uniformizar o tratamento de tais dispositivos quando usados com os pacotes gráficos citados anteriormente.

Na seção seguinte, são analisadas as características e a natureza das informações envolvidas em aplicações PAC, constituindo o tema central do presente trabalho.

3.3. Modelagem de Dados PAC

3.3.1. Análise Geral

Convencionalmente, um sistema PAC pode ser visualizado como uma cadeia de programas ou módulos de aplicação que tratam diversos arquivos de entrada e produzem vários arquivos de saída. Estes arquivos são na realidade diferentes representações de um mesmo objeto de projeto, de modo que o formato dos arquivos difere de um para outro, de acordo com o módulo que os utiliza. Assim, são necessárias facilidades de transformação de formatos quando se pretende conectar os módulos entre si.

Cabe aqui definir projeto de um sistema físico como sendo a criação de um modelo, suficientemente detalhado, que permita a construção/fabricação/montagem de tal sistema dentro de parâmetros de performance previamente estabelecidos. Ao modelo detalhado que se pretende obter, denominar-se-á objeto de projeto. O processo de projeto é definido, portanto, como sendo o conjunto de ações tomadas durante a criação de um objeto de projeto.

Duas formas de tratamento do esquema descritivo que representa o objeto de projeto são possíveis: através dos programas de aplicação ou através de sistemas de gerência de dados. No primeiro caso, a representação do esquema é tratada diretamente pelo programa de aplicação, através da declaração de variáveis e criação/gerência direta de arquivos, correspondentes aos objetos e associações que definem o objeto de projeto. Esta opção é indesejável, pois a dependência dos dados é muito alta o que torna complexa a gerência de dados de projeto, cuja estrutura pode mudar frequentemente.

Além disso, a redundância de dados causa problemas na manutenção da consistência dos mesmos. Assim, quanto mais os objetos de projeto crescem em complexidade (aumentando assim a quantidade de dados), a deficiência de sistemas que não ajudam o projetista na manutenção da consistência se torna evidente, razão pela qual tem sido dedicado grande esforço ao estudo da integração de sistemas de gerência de banco de dados em sistemas PAC (/EBER 82/, /EAST 82/, /LILLE 82/, /LORIE 82/).

O uso de Sistemas de Gerência de Banco de Dados abstrai do programa de aplicação o problema de gerência/criação de arquivos e variáveis que correspondem ao esquema de projeto, trazendo diversas vantagens, tais como: redução da redundância de dados e da representação de esquemas, melhor controle da consistência e integridade dos dados, isolamento do programa de aplicação dos problemas de administração de dados e de espaço de armazenamento.

Outro ponto referente à representação da Informação em aplicações PAC, diz respeito à caracterização dos seguintes tipos de Informação (/ENCA 83/):

- * Informações que constituem uma descrição primária do projeto;
- * Informações que são derivadas dos dados primários, e que podem ser chamadas de informações secundárias;
- * outras informações que constituem uma descrição adicional do projeto (por exemplo, dados de fabricação).

Desse modo, tem-se que os dados secundários levam a problemas de consistência durante alguma operação de atualização dos dados. Se os dados primários são mudados, os secundários devem ser invalidados.

Ainda dentro deste aspecto, é importante salientar que em aplicações PAC não é rara a necessidade de se definir múltiplos "pontos de vista" ou "visões" dos dados, pois o ambiente de projeto consiste de múltiplas aplicações que acessam os mesmos dados sob diferentes perspectivas. Por exemplo, o que é visto como um único objeto em uma aplicação pode ser visualizado em outra como uma reunião estruturada de objetos mais atômicos. Além disso (ou em vista disso), deve ser possível mapear a definição de um objeto composto em uma visão sobre as definições dos objetos componentes da outra visão, de tal modo que as operações na primeira sejam mapeadas em operações na segunda (/JOHNS 83/).

Nas próximas seções serão analisados os principais aspectos de estruturação de dados em sistemas PAC. Estes aspectos serão divididos em dois grupos: aspectos micro e aspectos macro, caracterizando os tipos de informação coexistentes em um sistema PAC.

3.3.2. O Aspecto Micro de PAC

Neste item serão analisadas as características de PAC que envolvem os aspectos ligados ao tratamento das informações de um objeto de projeto. Este tratamento se preocupa com o desenvolvimento do objeto de projeto de forma que determinadas propriedades e características pré-estabelecidas para aquele objeto sejam garantidas. Tais aspectos são aqui denominados de aspectos micro de PAC, posto que envolvem as informações de um determinado objeto de projeto, sem se preocupar com os interrelacionamentos de tais informações com informações de outros objetos de projeto, que coexistem em um sistema PAC.

Dentro da perspectiva micro, um objeto de projeto é definido em termos de objetos e associações elementares que definem sua estrutura, e regras de formação que devem ser obedecidas por estes elementos e que são definidas a partir das propriedades pré-estabelecidas para o objeto de projeto.

Sob o ponto de vista micro, o que importa de fato é o tratamento propriamente dito do objeto de projeto, isto é, a atribuição de valores ou instâncias (dados operacionais) aos elementos que

compõem o objeto de projeto. Ao atribuir tais valores, o projetista segue regras heurísticas, baseadas em seu conhecimento particular sobre o objeto de projeto, enquanto o sistema PAC se encarrega de verificar as regras de formação que foram definidas para o objeto. Este conjunto de valores forma assim uma versão, que pode ser entendida como uma alternativa atual de projeto do objeto em questão. Eventualmente, pode-se estar interessado em novas alternativas de projeto, o que leva à geração de novas versões.

Evidentemente, a criação de um objeto de projeto nos termos descritos acima não pode ser feita isoladamente. Este processo de criação envolve, eventualmente, outras bases de dados que não a do próprio objeto, o que leva ao fato de que a criação de um objeto pode ter reflexos em outras bases de dados e, conseqüentemente, nos processos de projeto que atuam sobre estas bases. Os aspectos envolvidos neste tipo de problema são tratados a seguir.

3.3.3. O Aspecto Macro de PAC

Na abordagem dos aspectos micro de PAC não se está interessado nos reflexos que as informações de um objeto de projeto têm sobre as informações de outros objetos que coexistem em um ambiente PAC, ou sobre a própria concepção do objeto em questão. Os aspectos macro de PAC procuram tratar estes elementos, ou seja, tratar a sistemática de projeto. Na abordagem dos aspectos macro, está-se preocupado com o próprio processo de projeto: evolução do projeto; evolução da concepção do objeto de projeto; interrelacionamentos entre vários objetos de projeto e, portanto,

entre os vários processos de projeto; etc.

Conforme será discutido a seguir (figura 3.6), a atividade de projeto é inerentemente multi-nível, ou seja, envolve generalizações e agregações de conceitos, desde uma especificação de grandes linhas até a obtenção de um produto acabado, envolvendo atividades de projeto "top-down" e "bottom-up" sobre a hierarquia de visões (ou facetas) que representa o objeto de projeto, sobre o qual atua o processo de projeto.

Enquanto no aspecto micro está-se predominantemente preocupado com cada faceta, no aspecto macro está-se preocupado com o interrelacionamento entre facetas. Assim a modificação de especificações deve ser administrada de forma a não provocar inconsistências.

Apesar de sua importância, ao aspecto de gerência de projeto tem sido dada pouca atenção quando da modelagem de dados de projeto. A visão a ser aqui apresentada (capítulo 4) é integrar os aspectos micro e macro, permitindo um tratamento uniforme e consistente de ambos no contexto de PAC.

3.3.4. Modelagem do Processo de Projeto

Para se tratar adequadamente os aspectos micro e macro de PAC e para aumentar a utilidade e generalidade de sistemas PAC, é necessário um conhecimento profundo do processo de projeto, de modo que se possa obter uma metodologia que seja usada em tais sistemas.

Os principais objetivos para uma metodologia de projeto que possa ser aplicada a sistemas PAC são os seguintes (/EDER 82/):

- * a metodologia deve levar a definições precisas, de modo que o processo de projeto possa ser detalhado o suficiente para oferecer uma estrutura lógica adequada para uma implementação;
- * a metodologia deve ser estruturada e adaptável. Um processo de projeto específico deve ser obtido pela restrição dos parâmetros do modelo global do processo;
- * a metodologia deve levar em conta a "personalização", conforme o conhecimento básico do usuário. Não deve impor restrições inadequadas sobre a sequência de decisões, o que limitaria a criatividade;
- * a metodologia deve ser útil como um modelo de interação com o usuário. A interação entre usuário e sistema deve ser feita de acordo com o estabelecido pela metodologia, isto é, o sistema deve forçar certas sequências e restrições inerentes à metodologia;
- * a metodologia deve oferecer duas visões diferentes. Uma lógica, a visão funcional para o projetista, mostrando o estado de especificação e dependências. A outra é a visão de recursos, conveniente para a gerência do projeto, isto é, identificação da atual fase do projeto através do estabelecimento de

tarefas de projeto e metas para o potencial humano, estimativas de tempo e orçamento.

O que se apresenta a seguir é uma formalização do processo de projeto, que tenta levar em consideração alguns dos elementos expostos acima.

Pode-se dizer que o modelo de um determinado produto, submetido a um processo de projeto, pode ser abstraído do mundo real por uma sequência de duas operações: percepção e seleção, como nos mostra a figura 3.3. Percebe-se apenas um subconjunto da realidade, a qual podemos tratar através da associação de nomes aos objetos e às características e propriedades destes objetos. Como não se está, de modo geral, interessado em todos os aspectos da realidade perceptível, seleciona-se apenas aqueles aspectos que são relevantes à atividade de projeto. Esta seleção gera o que se pode chamar de "universo de discurso", que não é em si um modelo formal mas um modelo conceitual intuitivo dos objetos a serem projetados.

A partir deste modelo conceitual, dá-se início efetivo ao processo de projeto, que consiste numa sequência de estágios que se inicia com a percepção das necessidades do produto e termina com uma descrição final detalhada da configuração do mesmo. Cada estágio pode ser, em si, um processo de projeto ou ser decomposto em vários processos.

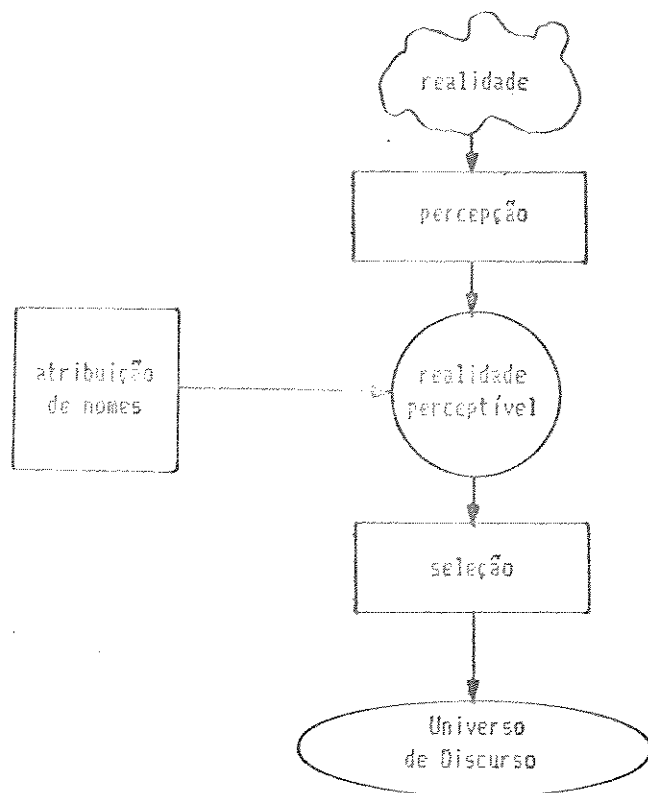


Figura 3.3. Modelagem do mundo real

Tomando os conceitos de /EBER 82/, pode-se imaginar o processo de projeto como sendo um processo ambientado em um espaço multidimensional, com componentes cuja combinação estabelece a direção seguida durante a evolução de um objeto de projeto. Estas componentes são (figura 3.4):

1. Percepção e Definição do Problema: esta componente do processo compreende passos que procuram estabelecer as necessidades do produto, tais como: análise de demanda, identificação dos objetivos e problemas, análise do estado técnico na área de engenharia, estudo de viabilidade, classificação dos requisitos e atribuição de prioridades, revisão, estabelecimento e documentação dos requisitos estabelecidos. Em fases posteriores do projeto, são

detectados subproblemas, de modo que mudanças nos requisitos do produto podem ser consideradas.

2. **Abstração/Concretização:** devido à impossibilidade de derivar diretamente a descrição de um sistema a partir dos requisitos, deve ser colocado em prática um processo de resolução e síntese progressiva. Durante o projeto, certos aspectos são abstraídos, sendo que o objeto resultante desta abstração é subsequentemente refinado. Isto é feito decompondo-se a estrutura funcional do objeto, para que em seguida o processo evolua sobre cada elemento morfológico resultante da decomposição.

3. **Modularização:** esta componente envolve um processo que evolui em paralelo com o processo de abstração/concretização, tendo como meta a modularização do produto. A complexidade de um objeto de projeto é determinada pelas suas propriedades, de forma que o objeto é sucessivamente refinado, através do estabelecimento de estruturas de baixo nível, ou seja, de módulos primitivos.

4. **Representação e Modelagem:** em qualquer fase de projeto, é necessário que os objetos de projeto tenham: modelos matemáticos específicos, notações e descrições, e representações gráficas associadas (e.g., simulações discretas ou contínuas, modelos de elementos finitos, diagramas, desenhos, protótipos, etc.).

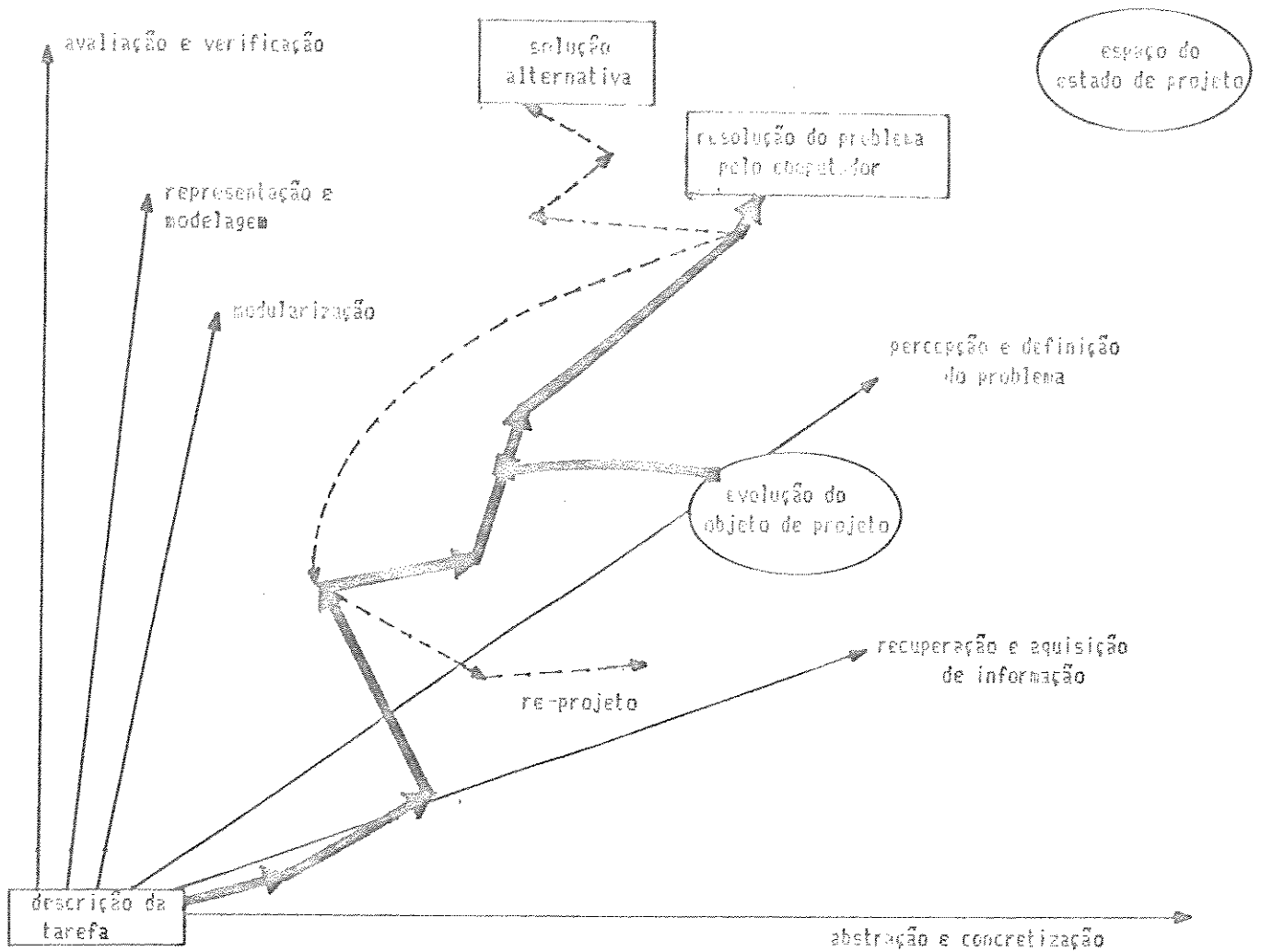


Figura 3.4. Espaço n-dimensional do Processo de Projeto (/ERER B2/)

5. **Avaliação e Verificação:** o objeto de projeto passa por numerosos procedimentos de avaliação e verificação. Os resultados destas verificações determinam se deve ou não haver repetição de passos de projeto. Tais ciclos de realimentação são típicos para todas as fases do processo de projeto e devem ser planejados adiantadamente juntamente com reavaliações dos requisitos.

6. **Aquisição e Recuperação da Informação:** a comunicação e troca de informação constitui a maior parte do trabalho de projeto.

A automação de tarefas de projeto certamente vai intensificar esta situação.

A seguir procura-se estabelecer um modelo mais rigoroso e formal, para o processo descrito acima, que sirva como ponto de partida para a ambientação do processo de projeto em um sistema computacional. Um modelo devidamente refinado para este processo pode ser visto na figura 3.5, modelo este que procura conter componentes em todas as direções do "espaço de projeto" mostrado na figura 3.4.

Neste modelo, o ciclo interno opera sobre uma especificação de projeto e consiste principalmente das operações de síntese, análise e avaliação. Aqui, a síntese consiste na coleta e organização de especificações providas pelo nível superior de projeto e no delimitamento do problema em si. Esta etapa é seguida pela geração de soluções candidatas, constituindo a atividade característica do processo de projeto.

A partir daí, as soluções são analisadas para se determinar as características que são relevantes às especificações existentes, o que é feito por cálculo, simulação, construção de modelos e outros recursos que auxiliarão a execução desta etapa. Usando os resultados da análise, as soluções candidatas são avaliadas com base nas especificações adotadas. O desvio do projeto preliminar em relação a especificação é então utilizado na operação de síntese. O segundo ciclo fecha-se com a apresentação dos resultados, já de acordo com a especificação do projeto, e com o retorno ao processo do nível

superior.

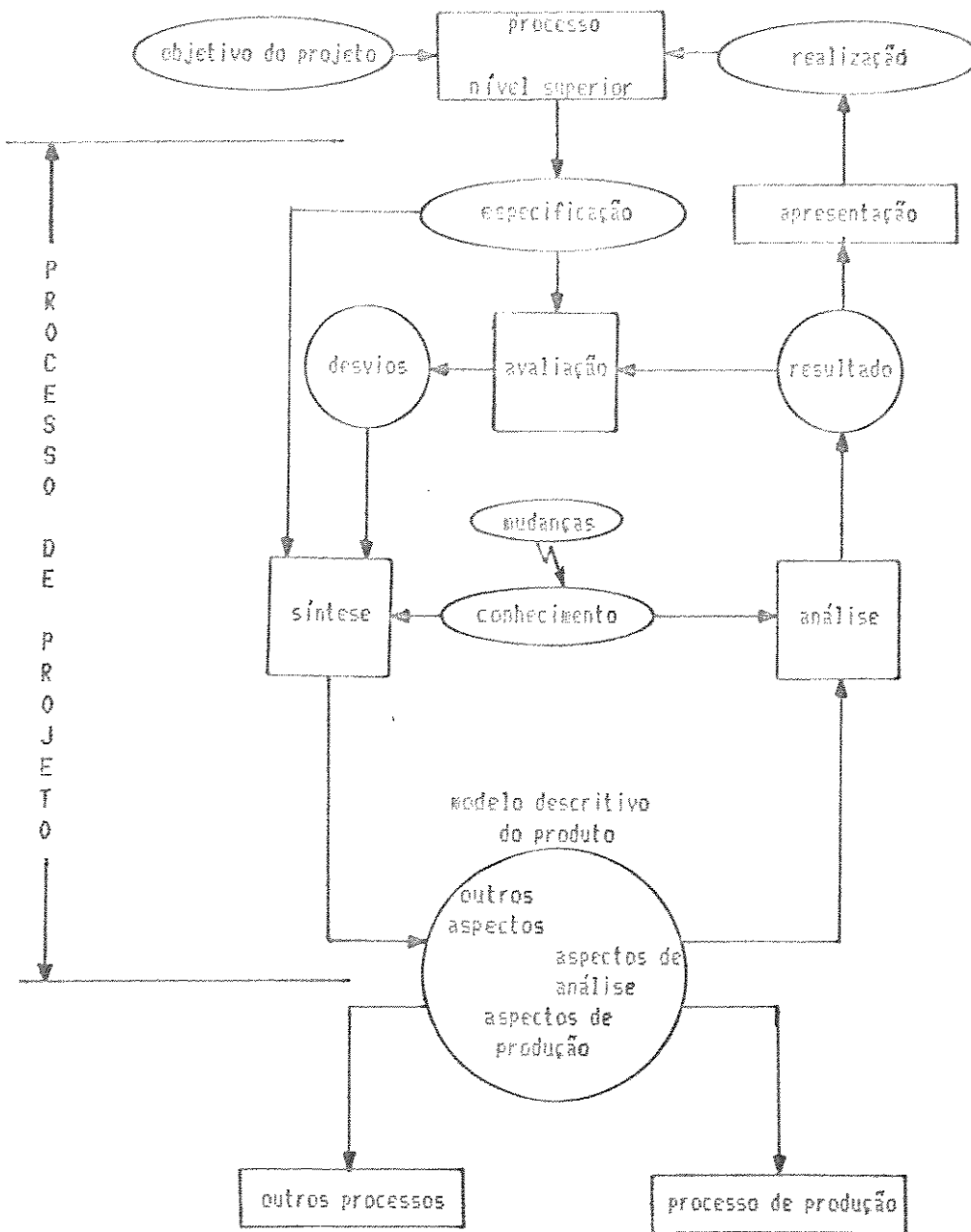


Figura 3.5. Modelo Refinado do Processo de Projeto (/ENCA 83/)

Devem ser notados alguns aspectos importantes no modelo. O processo de projeto não apenas gera a informação necessária para a produção do produto, mas também a informação necessária para a

operação de análise do processo de projeto e para todos os outros processos que o seguem. Este modelo descritivo, que é gerado pela síntese e comparado com a especificação via análise e avaliação, torna-se o ponto central de todas as operações seguintes.

Outro aspecto importante neste modelo é o fato de que o conhecimento disponível não é necessariamente fixo. Seria intolerável se, em esquemas de projeto e de produção de larga escala, o início do processo de projeto fosse adiado até que todo o conhecimento necessário estivesse disponível. Assim, o processo começa com uma pequena gama de conhecimentos que são renovados em cada iteração e/ou por outros processos.

A figura 3.6 representa de uma outra forma o processo de projeto. De fato, este processo consiste de uma cooperação hierárquica entre vários processos e subprocessos. Em cada um dos níveis, o processo de projeto descrito anteriormente se insere dentro de um processo de ambientação, no qual se procura, basicamente, criar, coordenar e gerenciar o processo e os recursos (e.g., conhecimento) necessários ao processo de projeto.

Para finalizar, deve-se atentar ao fato de que um processo de projeto pode ser formulado em termos de regras, o que permitiria a ambientação do processo em computadores. Contudo, embora o homem, como processador de informações, possa trabalhar com regras de modo imediato, esta é uma tarefa difícil para um computador, no que se refere ao reconhecimento de situações (síntese). Como os métodos computacionais de inteligência artificial e reconhecimento de

padrões estão ainda em fase de pesquisa, atualmente os métodos de PAC geralmente dividem o processo de projeto em duas atividades principais: síntese, que é associada ao fator humano, e análise, que é associada ao computador.

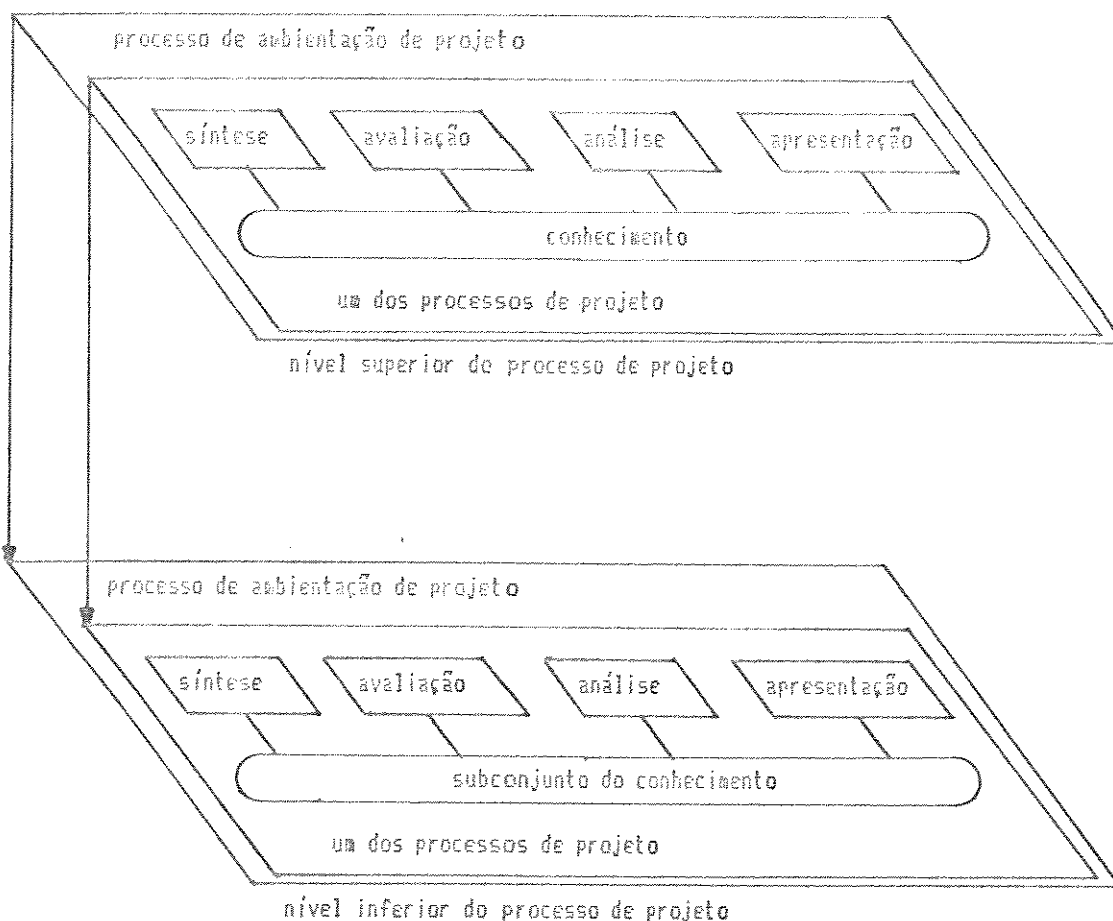


Figura 3.6. Cooperação entre processos de projeto

3.3.5. O Processo de Projeto em Ambiente PAC

Todo processo de projeto possui um tempo de vida, durante o qual ele interage com outros processos, criando-os e/ou comunicando-se com eles. Durante este tempo de vida, o processo pode alternar períodos de atividade e de inatividade, de modo que em determinado

ponto do tempo de vida, podem existir inconsistências no projeto de diversas naturezas (e.g., documentação incompleta, desvios em relação aos objetivos, etc.).

Dessa forma, o estado de um processo de projeto é determinado e representado pelo conhecimento de certos elementos que se estabelecem durante este processo. Estes elementos (desenhos de projeto, documentação, conhecimento técnico, etc.) constituem os recursos específicos do processo, recursos estes que devem ser reservados para uso exclusivo deste processo.

A mesma situação existe em um processo de projeto ambientado em um sistema PAC, aqui denominado processo PAC. Neste caso, a representação do estado de um processo PAC está relacionada com o armazenamento da Informação (estruturas de dados e sua representação) e com o conhecimento do operador sobre o processo PAC (projetista).

O armazenamento da Informação pode ser analisado sob diferentes perspectivas. Os aspectos de hardware, por exemplo, são de menor interesse ao usuário final de PAC. Por outro lado, são importantes os problemas relacionados com a manipulação e representação do estado do processo a nível de sistema operacional, base de dados, sistema de gerência de arquivos, e a nível de linguagem de programação. Nestes níveis, os objetos na base de dados, a sequência de valores em um arquivo ou os valores num programa juntamente com os esquemas correspondentes representam uma parte do estado do processo PAC em um dado instante.

A segunda parte da representação do estado do processo PAC é, como se disse, o conhecimento associado existente na mente do projetista auxiliado por computador. Assim, importantes tarefas em PAC são:

- * comunicar o esquema do estado do processo PAC ao computador de uma forma compreensível à máquina;
- * completar uma instância deste esquema com valores em um processo PAC particular.

Dessa forma, é importante que a estrutura da informação seja bem definida e especificada de acordo com a aplicação a que se destina um sistema PAC. Devido à incapacidade do computador em realizar a tarefa de síntese no processo de projeto, o desenvolvimento de um modelo - ou mais precisamente: o conjunto de modelos que pertencem ao escopo de um sistema PAC - geralmente é uma tarefa a ser executada pelo projetista do sistema PAC. O usuário do sistema pode desenvolver modelos de produtos somente dentro das restrições impostas pelo projetista do sistema PAC.

A descrição do conjunto completo de modelos possíveis de serem desenvolvidos pelo usuário do sistema PAC é chamado esquema descritivo. O desenvolvimento do esquema é uma tarefa que não pode ser realizada pelo simples estudo antecipado das propriedades dos objetos a serem projetados. O esquema e seus vários subesquemas são significativamente influenciados pelas operações que serão executadas sobre o esquema, operações estas que correspondem às ações tomadas pelos processos de projeto que estarão associados à

aplicação PAC.

Assim, o critério de avaliação e análise de um esquema não é sua correção mas sim sua adequação, no sentido de que um esquema deve ser adequado a sofrer transformações de e para vários subesquemas, requeridas pelos diferentes passos de projeto. Esta transformação de esquemas é parte fundamental do processo PAC, visto que durante o processo, a concepção inicial de um projeto pode sofrer alterações, o que implica na mudança dos modelos conceitual e descritivo, e portanto do esquema.

Basicamente, a informação sobre um objeto a ser projetado e que deve ser fornecida e/ou desenvolvida no sistema PAC contém informações estruturais, geométricas e de fabricação. Além disso é importante poder-se representar a informação relacionada ao processo de projeto como um todo, de forma consistente com os aspectos já mencionados. Para que o computador possa manipular ambas as informações, o desenvolvimento do esquema deve obedecer aos seguintes requisitos:

- * as entidades (objetos) e seus atributos (características e propriedades dos objetos), relativos a um determinado projeto devem ter nomes atribuídos a elas, de tal modo que sejam únicos dentro do ambiente em que são usados;
- * para facilitar as tarefas de consultas ou de variação das características do objeto, é conveniente dividir o esquema em subesquemas, de tal forma que cada um contenha dados na forma mais

adequada para a ação a ser tomada dentro deste subesquema:

- * estabelecer meios para transformação dos dados dos subesquemas de modo a possibilitar a apresentação destes dados em uma forma inteligível ao usuário.

A modelagem em PAC, a nível de projetista do sistema propriamente dito, consiste em aliar o planejamento do esquema descritivo ao planejamento do processo de projeto, o que é feito através das seguintes tarefas, que devem ser executadas paralela e coordenadamente:

a) planejamento do esquema:

- identificação dos objetos a serem tratados,
- especificação das associações entre os objetos,
- especificação dos atributos dos objetos,
- especificação da faixa de valores permitida a estes atributos,
- introdução de subesquemas:

b) planejamento do processo de projeto:

- identificação dos subprocessos,
- especificação dos subesquemas necessários aos subprocessos,
- especificação das precedências entre subprocessos,
- especificação dos requisitos de validade de dados para cada subprocesso:

c) planejamento da linguagem de comunicação:

- especificação dos pacotes de informação para comunicação homem/máquina,

- especificação das regras para formulação e representação destes pacotes.

Uma vez executadas estas tarefas, tem-se a percepção de quais são os processos a serem executados no ambiente PAC: quais objetos de projeto serão submetidos a estes processos; quais informações devem fluir de um processo a outro; e quais processos têm reflexos sobre outros processos. Isto estabelece a conexão entre os aspectos micro e macro de um Processo PAC: ações tomadas a nível micro podem, eventualmente, exigir ações a nível macro, e vice-versa.

O próximo capítulo apresenta o MER/PAC, um modelo de dados definido a partir das necessidades de projeto comentadas neste capítulo. Este modelo procura fornecer primitivas que permitem representar os aspectos micro e macro com uma mesma ferramenta de modelagem. Esta intenção se justifica pelo fato de que o que se convencionou identificar de aspecto micro e aspecto macro de uma aplicação PAC representa, na verdade, visões complementares de etapas de um processo de projeto.

Como visto na seção 3.3.4, se o processo de projeto pode ser representado como uma hierarquia de processos que atuam sobre diferentes bases de dados, tem-se que os aspectos que são visualizados como macro por um processo são vistos como aspectos micro pelo processo hierarquicamente superior. Assim, ao se modelar os dados de um determinado processo tem-se a necessidade de modelar dois aspectos: as informações referentes ao processo em si (aspecto

micro) e as informações importantes para o interrelacionamento deste processo com outros processos na hierarquia de projeto (aspecto macro), o que recomenda o uso de uma única ferramenta para ambas as modelagens.

CAPÍTULO 4. O MODELO ENTIDADE-RELACIONAMENTO PARA PAC

4.1. Introdução

A principal característica de aplicações PAC é a complexidade das informações envolvidas e do controle do processo de projeto e, portanto, do controle do fluxo das informações entre os diversos níveis deste processo.

Em face disto, a utilização de Sistemas de Banco de Dados em sistemas PAC vem se tornando, nos últimos anos, objeto de intensa pesquisa. Além das vantagens gerais decorrentes do uso de Sistemas de Banco de Dados, existem outras razões que justificam e usam de tais sistemas em sistemas PAC.

A principal delas é que o sistema de banco de dados age como elemento de comunicação e integração, ligando todas as atividades PAC, distribuindo os dados relevantes para as atividades correspondentes e fornecendo meios para a realimentação de informações entre as diversas atividades, facilitando o controle dos dados dos diferentes processos de projeto.

Em PAC, o volume de manipulações das informações é muito grande, o que reforça o uso de SGBD's para o controle da redundância, consistência e integridade dos dados, garantindo a qualidade da informação armazenada no sistema PAC. Além disso, o processo de projeto pode evoluir adequadamente, pois o SGBD gerencia

o acesso aos dados por vários programas, que em última análise representam os diversos níveis de um processo de projeto que coexistem em um sistema PAC. Se o SGBD suporta um modelo de dados suficientemente poderoso, um alto grau de independência de dados é obtido, de forma que a estrutura da informação ou os programas de aplicação podem evoluir sem maiores impactos sobre o sistema geral.

Resumindo, o Sistema de Banco de Dados (SBD) se adequa otimamente ao papel de elemento de coordenação do sistema PAC, pois todas as informações pertinentes às atividades e ao fluxo de dados do sistema PAC podem ser armazenadas no banco de dados. Estas informações são atualizadas à medida que o processo de projeto evolui, o que assegura um trabalho coordenado de todos os processos residentes no sistema PAC.

Finalmente, um SBD assegura aos projetistas e engenheiros responsáveis pela fabricação e pela produção um acesso mais rápido e eficiente a informações de interesse às suas respectivas atividades (e.g. manutenção de múltiplas versões de um projeto, acesso a dados técnicos para fabricação de um produto, controle de dados da produção, etc.).

4.2. Sistemas de Banco de Dados para PAC

Uma vez que o desenvolvimento de um sistema de banco de dados representa um grande esforço e já que o uso de tecnologia de banco de dados surge como necessidade em um grande número de aplicações de engenharia, tenta-se fazer uso de pacotes comerciais

disponíveis no mercado. Infelizmente, estes sistemas falham em suportar aplicações de engenharia.

A razão disto é que estes sistemas não foram projetados para propósitos de engenharia, mas para resolver problemas administrativos. As características de banco de dados são bem distintas nestes dois domínios, diferenças estas que são frequentemente mencionadas na literatura.

Um banco de dados administrativo contém poucos tipos de entidades com um grande número de ocorrências, ao contrário de bancos de dados de engenharia, que contém muitos tipos de entidades com poucas ocorrências. Paralelamente a isto, em engenharia existe a necessidade de modelar e manipular associações complexas, o que não é o caso em administração, onde as associações são bem mais simples.

Uma função importante em PAC é o modelamento geométrico de objetos, de modo que bancos de dados PAC devem ser capazes de manipular dados geométricos (linhas, círculos, polígonos, etc.). Isto leva à constatação de que tipos de dados convencionais (inteiros, caráter, etc.), não são suficientes para aplicações de engenharia, exigindo tipos de dados de estrutura mais complexa.

Além disso, recursos adicionais devem ser garantidos em banco de dados PAC de forma a garantir a integridade semântica e a consistência de dados, já que variantes múltiplas ou modificações do mesmo objeto podem ser armazenadas simultaneamente. Associado a isto, tem-se que em aplicações PAC é necessária a definição de

múltiplas visões (facetas) dos dados, pois o ambiente de projeto consiste de múltiplos processos que acessam os mesmos dados em diferentes estados de desenvolvimento.

De fato, um processo de projeto envolve basicamente o desenvolvimento de um esquema (estrutura de dados), de forma que este se modifica continuamente durante a vida do projeto, até transformar-se em um produto. Tal característica não é considerada em aplicações de administração, onde o esquema tem um tempo de vida relativamente longo.

Estas características, aliadas à dinâmica geral de um processo de projeto, têm consequências importantes no que diz respeito aos períodos em que um banco de dados PAC está inconsistente. De um modo geral, um banco de dados pode ser visto como uma coleção de entidades que podem assumir determinados valores, sendo que o conjunto de todos estes valores definem um estado do banco de dados. Além disso, tem-se associado um conjunto de restrições de consistência e integridade, de modo que um estado de banco de dados que satisfaz estas restrições é dito ser um estado consistente.

Uma transação pode então ser definida como uma sequência de ações sobre a base de dados que transfere esta de um estado consistente para outro estado consistente. Durante este processo de transferência de estado, a base de dados pode estar temporariamente em um estado inconsistente.

Assim, outra diferença fundamental entre um banco de dados convencional e um banco de dados PAC é a duração dos períodos de inconsistência. Um banco de dados convencional é inconsistente somente durante uma transação, que dura apenas alguns segundos. Um banco de dados PAC, pelo contrário, se torna consistente apenas no final do processo de projeto. Assim, como se verá a seguir, o conceito de transação em PAC necessita de um tratamento especial.

Ao final de uma transação convencional, as atualizações são validadas usando-se as regras de consistência. Somente atualizações que não violem as regras são colocadas à disposição do banco de dados. No entanto, ao final de uma transação PAC, duas classes de regras de consistência devem ser verificadas durante a validação dos dados: regras locais e regras globais.

Para compreender melhor esta última observação, deve-se recordar o que foi apresentado no capítulo 3 sobre os aspectos micro e macro de PAC. Dentro do aspecto micro, um esquema referente a um objeto de projeto é definido em termos de objetos, associações e regras de formação que devem ser obedecidas por estes elementos. Do ponto de vista macro, consideram-se os reflexos da atividade de projeto sobre os vários esquemas definidos no ambiente PAC. Assim, ao se planejar o processo global de projeto, definem-se todos os subprocessos e os esquemas associados a cada um, bem como as precedências entre os subprocessos.

Para representar tais aspectos a nível de banco de dados, tem-se que as regras de formação dos elementos de um objeto de

projeto (aspecto micro) são correspondentes a regras de integridade e consistência de um esquema de banco de dados, regras estas que são aqui denominadas de regras locais. Por outro lado, o processo de projeto que atua sobre um determinado esquema pode estar subordinado a um processo de nível superior, que pode ter outros subprocessos subordinados atuando sobre outros esquemas de banco de dados. Assim, tem-se um conjunto de representações de objetos de projeto (esquemas) que estão interrelacionados. O processo de nível superior segue, portanto, um determinado conjunto de regras de integridade e consistência de banco de dados que se referem a este interrelacionamento, regras estas que se denominam regras globais em relação aos diversos esquemas.

A transação PAC se divide, assim, em duas fases que evoluem paralelamente: transação micro, que verifica as regras locais, e transação macro, que verifica as regras globais para a validação dos dados de projeto. A transação macro não permite, contudo, que os dados invalidados globalmente sejam perdidos, pois isto não é admissível em um processo de projeto, onde os dados só atingem consistência ao final deste processo.

Em vista de todos estes elementos, tem-se que os requisitos específicos para banco de dados PAC e, portanto, para Modelo de Dados PAC, são baseados nos seguintes itens:

1. **Mecanismos de Construção de Estruturas de Dados:** em um projeto, um grande número de elementos de dados estruturados de vários tipos devem ser processados e armazenados (dados geométricos, desenhos,

texto, etc.). O Modelo de Dados deve fornecer aos usuários mecanismos flexíveis e poderosos para construção de estruturas de dados. Dentro deste aspecto, existem dois pontos de importância:

A) Tipos não-convencionais de dados: o sistema deve suportar tipos de dados mais complexos de interesse para PAC, adotando o conceito de tipo abstrato de dados (TAD). Um TAD é definido como um encapsulamento de uma estrutura de dados com um conjunto de operações/operadores associados a esta estrutura (/STON 83/, BERZ 83/). Pode-se considerar, por exemplo, um tipo inteiro como um TAD, cuja estrutura depende da máquina na qual ela é implementada, sendo definidos os operadores '+' (soma), '-' (subtração), '*' (multiplicação) e '/' (divisão), cujos detalhes de implementação não são visíveis a um procedimento externo.

B) Estruturas complexas de dados: como dito anteriormente, em aplicações PAC é comum a existência de relacionamentos complexos, muitas vezes com o objetivo de se modelar objetos complexos, isto é, objetos que são compostos por outros objetos. Dessa forma, um Modelo de Dados que se pretenda útil em PAC deve permitir a modelagem de relacionamentos dos seguintes tipos:

- * relacionamentos entre entidades;
- * relacionamentos entre entidades e relacionamentos;
- * relacionamentos entre relacionamentos.

Por outro lado, um Modelo de Dados PAC deve permitir dois tipos de definição de entidades e relacionamentos:

- * definição estrita, em que cada tipo de entidade/re relacionamento determina os nomes e os domínios de valores de seus atributos;
- * definição complexa, em que o tipo de enti-

dade/relacionamento consiste de uma combinação hierárquica de primitivas cuja ocorrência é mutuamente exclusiva (/GHAL 82/) e/ou de primitivas cuja ocorrência é simultânea (/LORIE 82/). Neste tipo de definição, os elementos que compõem a primitiva complexa podem ter alguns atributos em comum e cada elemento tem por sua vez seus atributos locais.

2. Suporte para Múltiplas Visões de Projeto: frequentemente, no processo de projeto, diversos projetistas trabalham sobre um mesmo projeto, gerando diferentes alternativas simultâneas, a fim de criar variantes ou para escolher a melhor alternativa. Além disso, o mesmo objeto pode ser visualizado sob diferentes aspectos, dependendo da formação do projetista que analisa o objeto (e.g, um projeto de edifício requer engenheiros civis, engenheiros eletricitas, etc.) (/EAST 81/).

3. Controle do Processo de Projeto: com as eventuais alterações dos dados durante um processo de projeto (aspecto micro de PAC), pode surgir a necessidade de se alterar as definições dos objetos e relacionamentos e, conseqüentemente, o próprio esquema concernente ao processo (aspecto macro de PAC). Além disso, como este processo de projeto pode fazer parte de um processo de nível superior (como visto no capítulo 3.3.3), este pode se ver obrigado a alterar seus dados e esquemas. Da mesma forma, como o processo de projeto pode gerar subprocessos, estes também podem vir a alterar os dados e esquemas associados. O SGBD deve, então, garantir meios de controlar

tais operações, permitindo assim que o projeto possa se desenvolver dentro de uma abordagem "top-down" ou de uma abordagem "bottom-up". Portanto, o controle do processo de projeto exige que o SGBD trate adequadamente as transações PAC: transações micro e transações macro.

4. **Compartilhamento da Administração de Banco de Dados entre ABD e Projetista:** o ABD é o responsável pela definição das estruturas de dados do banco de dados, mas os projetistas devem ser livres para organizar seus bancos de dados de projeto de acordo com as necessidades através de comandos da linguagem disponível. O ABD tem a responsabilidade de avaliar os requisitos conflitantes dos usuários, definindo mecanismos de segurança e controlando a coordenação e o estado das atividades que utilizam o banco de dados podendo-se dizer, de fato, que o ABD assume o papel de gerente de projeto.

5. **Nível de Performance:** para assegurar alta produtividade e evitar insatisfação dos usuários, o sistema deve ter mecanismos rápidos de acesso de dados. Por exemplo, desenhos devem ser extraídos rapidamente para serem mostrados na tela de um monitor de vídeo. Linguagens de consulta ("query languages") devem ser disponíveis, oferecendo facilidades para a produção de relatórios, análises estatísticas, etc e ao mesmo tempo apresentarem características de linguagens naturais.

6. **Controle de Integridade e Consistência:** o problema de integridade é o problema de assegurar a proteção da informação na base de dados

contra atualizações inválidas (/DATE 83/). Dessa forma, ferramentas para gerenciar a integridade e consistência de dados devem ser incorporadas ao SGBD, lembrando que a permissão da alteração de esquemas é um desafio a tal gerência. Além disso, é importante estabelecer de uma maneira clara as restrições concernentes aos relacionamentos e as referentes às ocorrências dos tipos de entidade. Tem-se assim dois tipos de restrições importantes:

A) **Restrições de Existência:** expressam as dependências de um tipo de entidade/reacionamento em relação a outro. Estas restrições podem expressar dependência de existência e imposição de existência.

B) **Restrições de Integridade:** dizem respeito às propriedades de um objeto ou associação entre estes, podendo ser estruturais, restrições de manipulação e restrições de formação de objetos. Tais regras serão abordadas na seção seguinte.

4.3. O MER/PAC

No item anterior, foram vistas as principais características que devem estar presentes em SGBD's e em modelos de dados que se proponham a ser úteis a aplicações PAC. Na presente seção, pretende-se propor um modelo de dados para aplicações PAC, levando em consideração estes aspectos.

Dentre os vários modelos de dados existentes, o Modelo Entidade-Relacionamento é um modelo adequado para a descrição de dados PAC, devido às características já vistas no capítulo 2. Assim, o MER permite:

- * definição de entidades (definição estrita);
- * definição de relacionamentos recursivos, n-ários;
- * definição de relacionamentos 1:N, M:N e 1:1;
- * definição de atributos multivalorados;
- * definição de domínios de valores;
- * explicitação da dependência de existência e de identificação entre entidades.

No entanto, a partir do que foi descrito na seção anterior, nota-se que o MER não satisfaz todas as exigências da aplicação PAC. Deste modo propõe-se a seguir uma extensão do MER, denominada MER/PAC, com as seguintes características adicionais:

- * definição de objetos e associações complexas;
- * definição de relacionamentos tanto entre entidades como entre relacionamentos;
- * utilização de tipos abstratos de dados (a nível organizacional);
- * explicitação de cláusulas de imposição de existência;
- * explicitação de dependência de existência entre entidades e relacionamentos, ou apenas entre relacionamentos.

Assim, o MER/PAC é definido através dos seguintes elementos:

- * entidade (regular/fraca);
- * relacionamento (regular/fraco);
- * agrupamento (regular/fraco);

* esquema (regular/fraco);

* definição de regras.

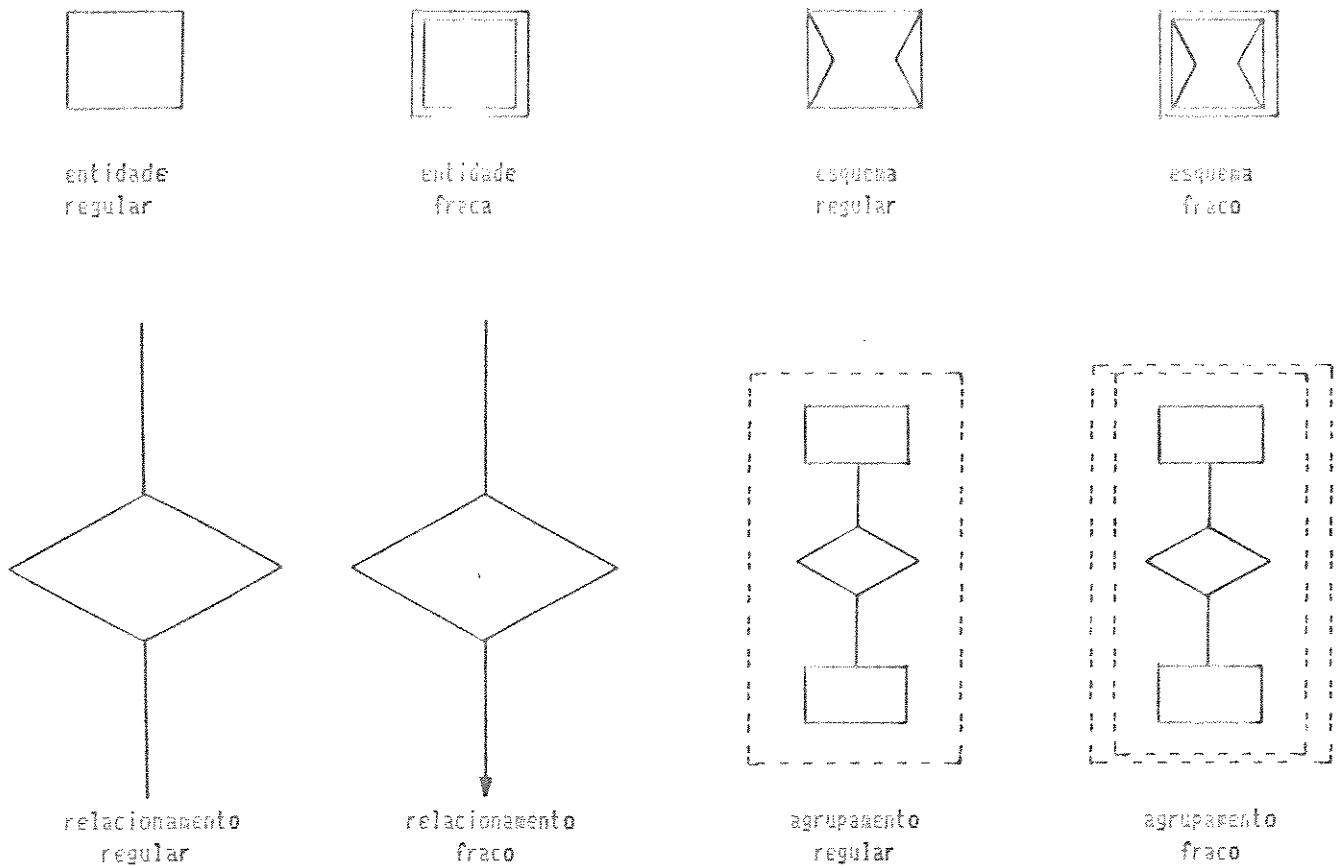


Figura 4.1. Primitivas MER/PAC

O MER/PAC é um modelo de dados que se situa no Nível de Abstração Descritivo, permitindo assim uma descrição formal mais próxima da visão abstraída do Mundo Real (Nível Conceitual). No MER/PAC os tipos de entidades e relacionamentos são classificados de acordo com dois critérios: complexidade de definição e tipos de dependência entre as primitivas. No que se refere à complexidade, as entidades e relacionamentos suportam dois tipos de definição, já

mencionados anteriormente (seção 4.2, item 1): definição escrita e definição complexa. Quanto ao aspecto de dependência, as primitivas podem ser classificadas como regulares ou fracas, conforme seja a dependência de existência destas primitivas em relação a outras. O que se procura frisar aqui é que, a nível de definição, cada classe de entidade tem uma classe correspondente de relacionamentos (e.g., um tipo de entidade fraca corresponde a um tipo de relacionamento fraco).

A figura 4.1 apresenta simbolicamente as primitivas de um diagrama MER/PAC. Como visto no capítulo 2 (item 2.5.2.2), muitos autores vêm propondo novos elementos e conceitos ao Modelo Entidade-Relacionamento. Em face destas propostas, este capítulo procura adaptar aqueles conceitos e elementos, propondo assim novas primitivas que servem aos propósitos de modelagem de PAC, e que serão expostas detalhadamente a seguir.

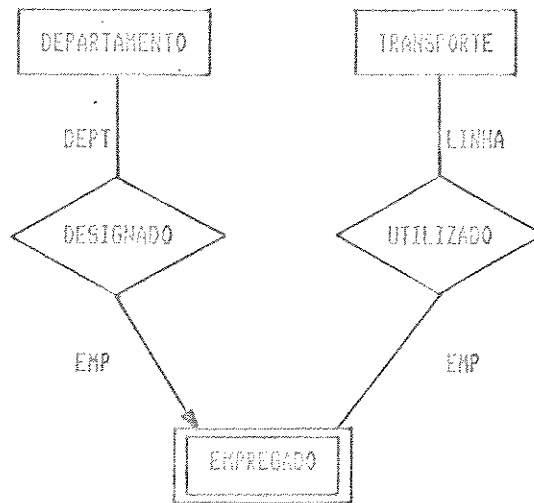
4.4. Primitivas do MER/PAC

A. ENTIDADES E RELACIONAMENTOS

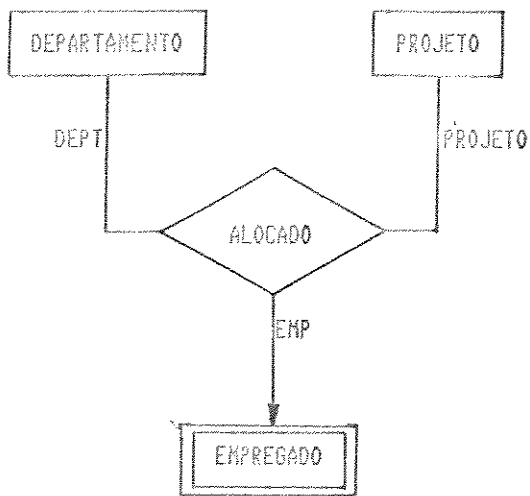
A primitiva entidade representa os objetos e fatos que têm existência e podem ser identificados no Mundo Real. As entidades que possuem propriedades comuns são classificadas em tipos de entidades. Assim, um tipo de entidade tem como elementos as entidades que são caracterizadas pelo mesmo grupo de predicados, denominados atributos. Por exemplo, carros podem ser caracterizados pelos atributos número de série, modelo, fabricante, cor, etc.

Existem entidades que têm sua existência na base de dados dependente da existência de outra entidade ou de um grupo de entidades, distinguindo-se os tipos de entidades em regulares e fracas. Uma entidade é dita ser regular se sua existência independe da existência de outras entidades e/ou relacionamentos. Caso a entidade não tenha estas propriedades, ela é denominada entidade fraca. Deve-se observar que o conceito de dependência de identificação (/CHEN 77/) não existe no MER/PAC. Isto porque não há no MER/PAC (um modelo de dados descritivo) o conceito de chave, o que quer dizer que não há necessidade de indicar os atributos que identificam uma entidade.

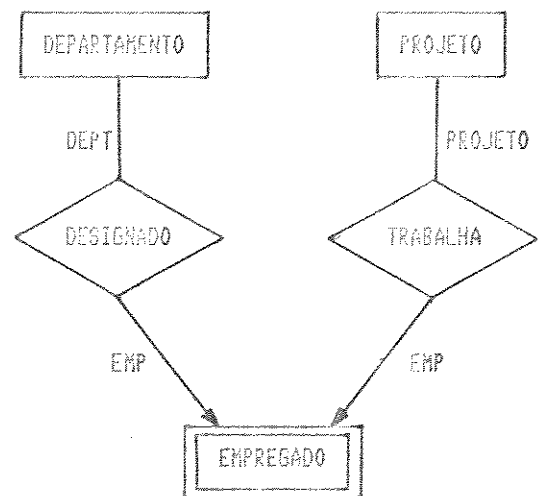
A primitiva relacionamento representa as associações entre os objetos (entidades) do mundo real. Um relacionamento regular representa aqueles relacionamentos que envolvam apenas entidades regulares. Da mesma forma, diz-se que um relacionamento é fraco se ele envolve alguma entidade fraca, de forma que um relacionamento que estabelece uma dependência de existência é um relacionamento fraco. As entidades envolvidas em um relacionamento exercem funções neste relacionamento, funções estas que se denominam papéis no MER/PAC (e MER). Na figura 4.2-a, DEPT, LINHA e EMP são os papéis que DEPARTAMENTO, TRANSPORTE e EMPREGADO exercem nos relacionamentos correspondentes. Note-se nesta figura que o relacionamento UTILIZADO é um relacionamento fraco, posto que envolve o tipo de entidade fraca EMPREGADO, embora não estabeleça a dependência de existência sobre esta entidade.



(A)



(B)



(C)

Figura 4.2. Entidades e Relacionamentos Fracos

É possível também estabelecer a dependência de existência de um tipo de entidade em relação a dois ou mais tipos de entidades. Isto pode ser feito de duas formas, ilustradas na figura 4.2-b e 4.2-c ("Um empregado deve estar alocado em um departamento e a um projeto"). Na figura 4.2-b, o tipo de relacionamento fraco ALOCADO

possui um único conjunto de atributos, referente à dependência do tipo de entidade fraca EMPREGADO em relação aos tipos de entidades DEPARTAMENTO e PROJETO. Já na figura 4.2-c tem-se representada a mesma dependência simultânea, mas de forma diferente, através dos dois relacionamentos fracos DESIGNADO e TRABALHO.

B. RELACIONAMENTO ENTRE ENTIDADES E/OU RELACIONAMENTOS

No MER/PAC é possível estabelecer um relacionamento envolvendo outros relacionamentos. Isto é feito através da abstração de um ou mais relacionamentos e entidades correlatas em um novo objeto, denominado agrupamento (figura 4.3). A cada agrupamento deve ser associado um conjunto de regras de formação de ocorrências, que define qual o conjunto de ocorrências das entidades/relacionamentos que definem uma ocorrência do agrupamento. Um agrupamento comporta os conceitos de agregação e generalização (/SMIT 77a/, /SMIT 77b/, /SMIT 80/) de primitivas, possuindo, para isto, regras de formação associadas e podendo ou não possuir atributos próprios.

Através de agrupamentos é possível definir objetos complexos e relacionamentos mutuamente exclusivos, conceitos que, como dito anteriormente, são importantes em PAC.

AGRUPAMENTO

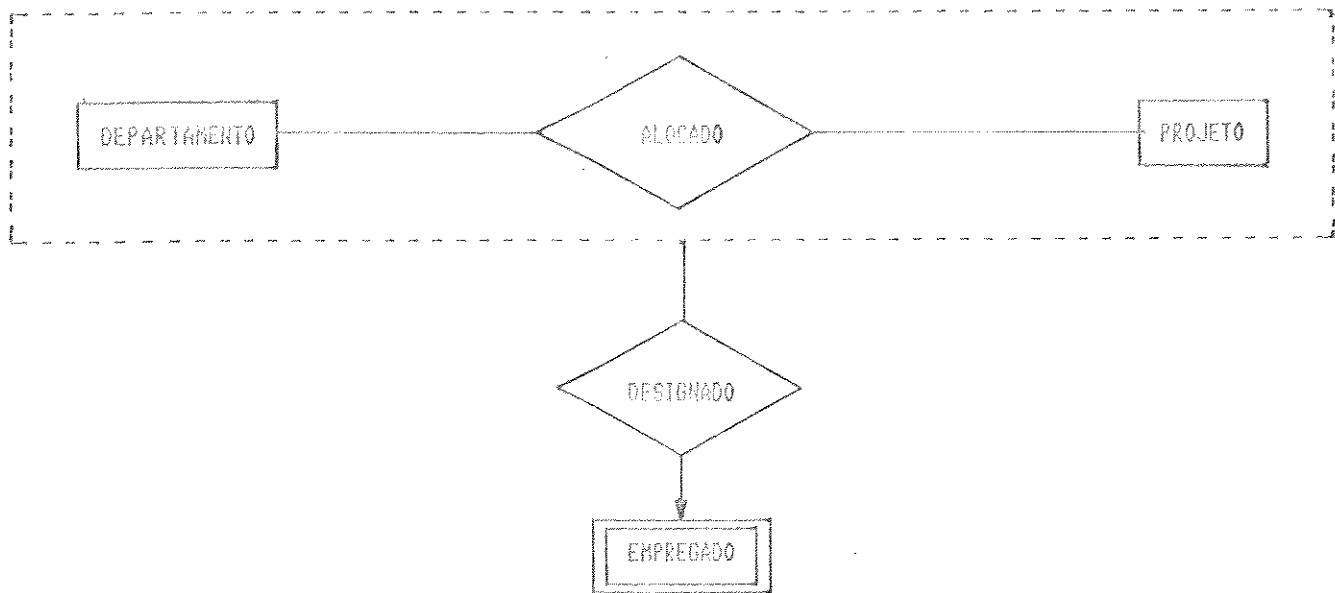


Figura 4.3. Agrupamento

B.1. Relacionamentos Mutuamente Exclusivos

Relacionamentos de tipos distintos são ditos serem mutuamente exclusivos (RME) se a sua existência for mutuamente exclusiva, isto é, a ocorrência de um tipo de relacionamento implica a não existência dos outros tipos de relacionamentos. A figura 4.4 mostra a representação de RME's utilizando a primitiva agrupamento.

No exemplo mostrado, as ocorrências possíveis para o agrupamento LECIONA são LEC_ENG_1 (LEO,TELEFONIA) ou (exclusivo) LEC_MED_1 (LEO,ANATOMIA), onde LEO, TELEFONIA, ANATOMIA, LEC_ENG_1 e LEC_MED_1 são, respectivamente, ocorrências dos tipos de entidade PROFESSOR, DISC_ENG e DISC_MED e dos tipos de relacionamento LEC_ENG e LEC_MED. Em outras palavras, se um professor lecionar disciplinas de engenharia, não pode lecionar disciplinas de medicina, e vice-

versa. As regras de formação do agrupamento definem a natureza mutuamente exclusiva dos relacionamentos LEC_ENG e LEC_MED.

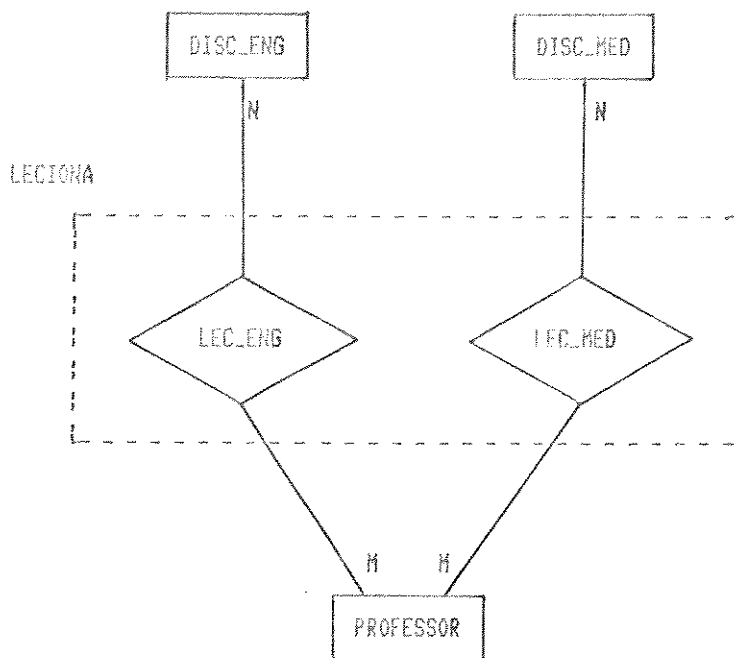


Figura 4.4. Relacionamento Mutuamente Exclusivo

B.2. Objetos Complexos

Um objeto complexo, representado na figura 4.5, consiste de uma composição de objetos simples associados a um objeto raiz. Os objetos subordinados podem ser complexos ou não e, semanticamente, representam uma descrição das características do objeto raiz. A ocorrência de um objeto complexo é na realidade a ocorrência de todos os objetos que o definem. É importante observar que uma ocorrência de um objeto complexo impõe que existam ocorrências de todos os objetos associados direta ou indiretamente ao objeto complexo. Assim, no caso do exemplo da figura 4.5, um carro MONZA SL é composto de RODAS CROMADAS com PNEUS RADIAIS e CALOTAS CROMADAS,

MOTOR TIPO GM1, CHASSIS TIPO 3, com EMBREAGEM ou (exclusivo) RODAS CROMADAS com PNEUS RADIAIS e CALOTAS CROMADAS, MOTOR TIPO GM1, CHASSIS TIPO 3, com CÂMBIO AUTOMÁTICO.

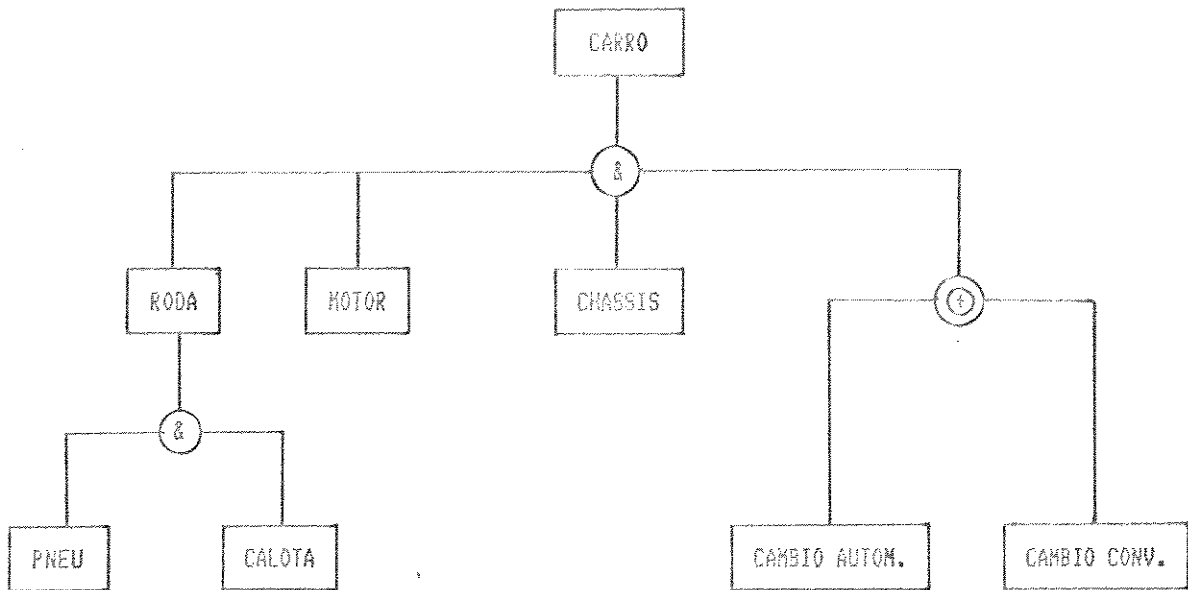


Figura 4.5. Objeto Complexo

Mais uma vez, para modelar um objeto complexo, utiliza-se o agrupamento (figura 4.6), onde as regras de formação do agrupamento CARRO_COMP devem garantir ocorrências das entidades RODA, PNEU, CALOTA, MOTOR, CHASSIS, CÂMBIO AUTOMÁTICO ou (exclusivo) CÂMBIO CONVENCIONAL, sempre que houver uma ocorrência da entidade CARRO. Deve-se ter em mente que estabelecer um relacionamento com uma entidade complexa, implica em estabelecer um relacionamento com todas as entidades associadas.

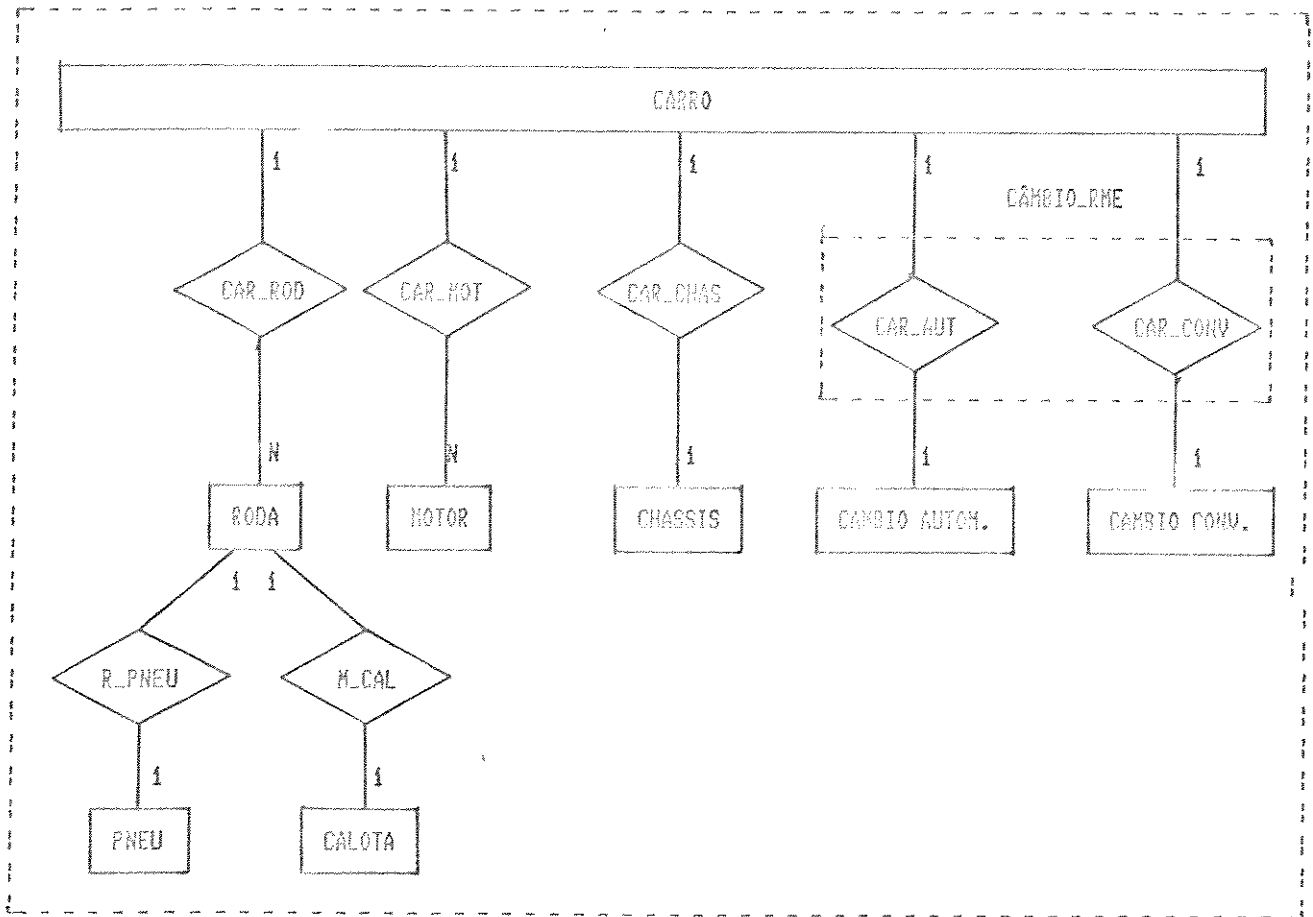


Figura 4.6. Agrupamento - Objeto Complexo

C. ESQUEMA

Um esquema consiste na definição de um conjunto de primitivas e cláusulas de integridade que constituem a descrição semântica da estrutura de dados de uma determinada aplicação. Desta forma, a primitiva esquema no MER/PAC representa um conjunto de entidades, relacionamentos e agrupamentos, que formam a estrutura de dados de uma determinada aplicação. Cada um destes conjuntos pode, por sua vez, sofrer alterações, tanto na estrutura de dados quanto

nos dados operacionais associados.

Assim, da mesma forma que entidades são classificadas em diferentes tipos de entidades, diz-se no MER/PAC que esquemas podem ser classificados em diferentes tipos de esquema. Dito de outra forma, no MER/PAC as instâncias de um tipo de esquema são denominados expansões, sendo que aos diversos conjuntos de dados operacionais associados a uma determinada expansão denomina-se versões da expansão ou simplesmente versões (figura 4.7). Cabe dizer que uma expansão pode por sua vez ser formada por um conjunto de tipos de esquema, da mesma forma que poderia ser definida por um conjunto de entidades e relacionamentos.

Com esta definição, o esquema é considerado também uma primitiva análoga a uma entidade, de forma que é possível descrever a estrutura global do banco de dados como sendo um conjunto de estruturas interrelacionadas. Assim, com esta primitiva é possível estruturar os dados de um processo de projeto, da forma descrita no capítulo 3.

Por outro lado, um tipo de esquema pode ter também seus atributos (representando, por exemplo, algumas especificações globais de projeto), sendo que devem existir dois atributos inerentes a esta primitiva: versão e expansão. O primeiro atributo mapeia do tipo de esquema para um domínio de valores adequado para a identificação de um conjunto de dados na base de dados. Já o atributo expansão faz o mapeamento do tipo de esquema para um domínio de valores adequado para a identificação da expansão à qual

está associada a versão indicada através do primeiro atributo.

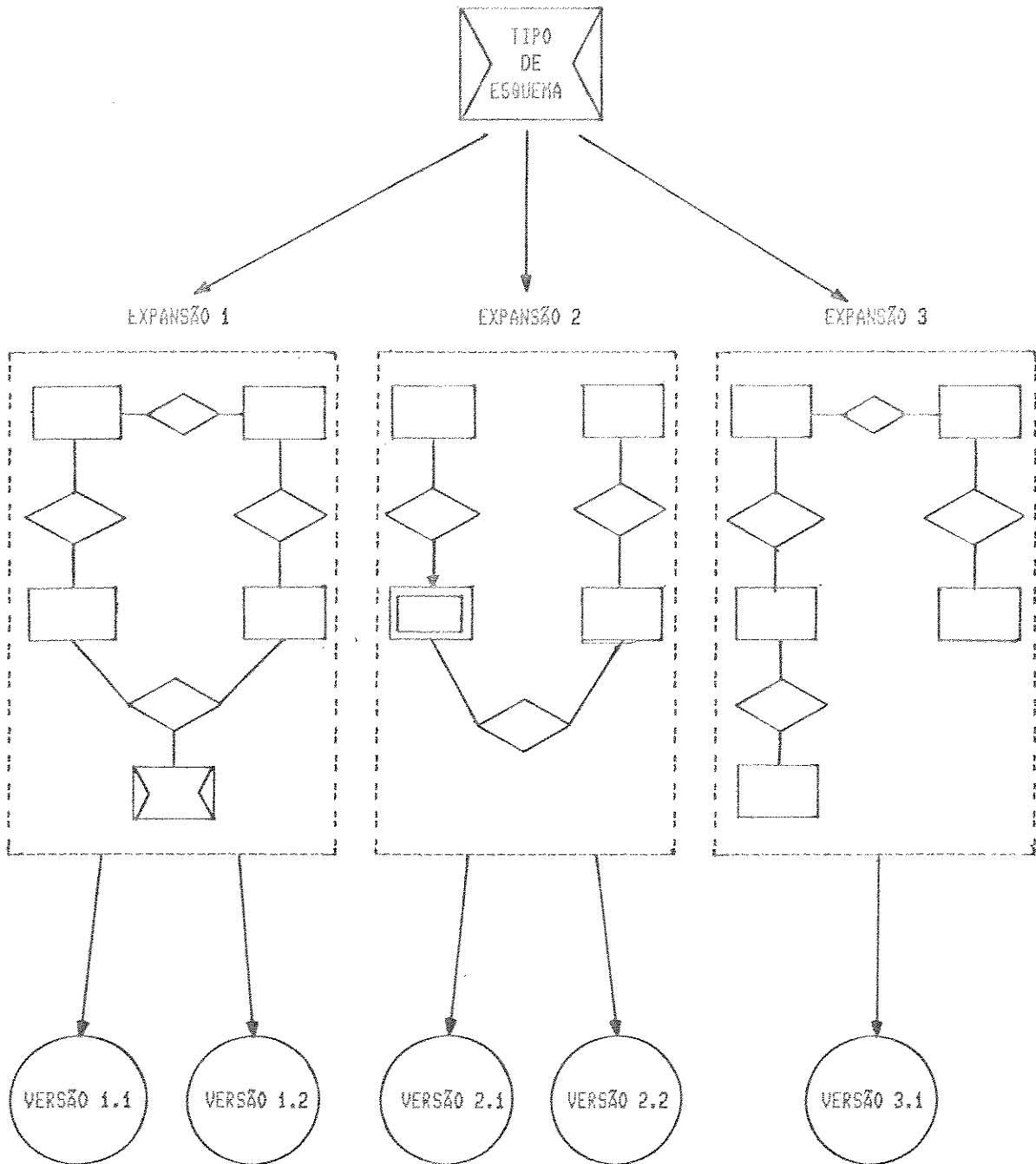


Figura 4.7. Primitiva Esquema do MER/PAC

De fato, o processo de projeto consiste basicamente de duas atividades:

1. alteração dos dados operacionais, que são analisados e comparados com as especificações de projeto. Isto gera conjuntos de soluções que refletem o progresso do projeto. Estes conjuntos correspondem às versões do MER/PAC;

2. eventual alteração da concepção do objeto de projeto, já que o processo de análise dos dados pode levar à revisão dos requisitos e características do projeto. Isto leva à alteração da própria estrutura dos dados (esquemas), criando assim "versões de esquemas" ou expansões para um mesmo objeto de projeto, versões estas que correspondem aos esquemas de um tipo de esquema.

O SGBD é utilizado pelos subprocessos para observar as regras de integridade de cada esquema (regras de integridade local) e pelo processo superior para verificar as regras que definem o interrelacionamento entre os esquemas (regras de integridade global) e para controlar os relacionamentos entre expansões e versões dos vários esquemas.

D. DOMÍNIO DE VALORES

Um Sistema de Gerência de Banco de Dados possui vários tipos de regras de integridade semântica, que definem com maior precisão os dados a serem armazenados na Base de Dados, situando-se entre estas regras a definição dos valores admissíveis no domínio de valores de certo atributo.

De fato, a definição do domínio de valores se insere em dois níveis de abstração. A nível descritivo (MER/PAC), simplesmente

se associa um tipo de entidade a um domínio de valores através de um atributo. O domínio de valores é identificado por um nome genérico qualquer, da mesma forma que é feito para atributos e tipos de entidade e relacionamento.

Já a nível organizacional, deve-se definir o tipo de dados do domínio de valores (inteiro, carácter, etc.), definindo-se também os valores admissíveis para o domínio. O que se propõe aqui é que os tipos de dados devem ser estendidos para tipos abstratos de dados (TAD), onde os tipos de dados convencionais são também consideradas TAD's. A partir daí, propõe-se também que o domínio de valores seja definido através de regras que especifiquem não só os valores admissíveis, mas também ações a serem tomadas caso estas regras sejam desobedecidas.

E. REGRAS: INTEGRIDADE, CONSISTÊNCIA E CHAVES

Dentro das primitivas MER/PAC definidas anteriormente, fica patente a importância da especificação de regras. De fato, a cada primitiva é possível associar um conjunto de regras que deve ser aplicado durante qualquer manipulação feita sobre aquela primitiva, mantendo assim a integridade e consistência requerida pela aplicação.

No MER/PAC as regras se dividem em três grupos importantes: regras de formação, regras de manipulação e regras estruturais. As regras de formação são aquelas que definem a forma como devem ser criadas e mantidas as ocorrências das primitivas MER/PAC. Estas

regras são as utilizadas na definição de agrupamentos, como visto no item B, sendo importantes na definição de objetos e relacionamentos complexos. As regras de formação são também importantes na definição de entidades, pois, não havendo no MER/PAC o conceito de chave, é através delas que se especificam as ocasiões em que não se permite a duplicação de ocorrências de tipos de entidades, a não existência de valores nulos, e índices.

As regras de manipulação são regras que devem ser verificadas durante as operações de inserção, remoção e alteração, podendo, entre outras coisas, expressar uma imposição de existência.

As regras estruturais são regras impostas sobre os valores de determinados atributos, especificando restrições sobre: valores admissíveis para o domínio de valores, valores permitidos para certos atributos, valores existentes na base de dados.

Uma outra aplicação importante de regras está no fato de que elas podem ser usadas tanto para definir regras locais que são verificadas a nível de transação micro, quanto para definir regras globais, situadas a nível de transação macro, como visto no item 4.2. Visualizando-se o problema de outra forma (através da figura 5.10), tem-se que as regras permitem um controle horizontal da integridade quando se referem às atividades locais de um processo de projeto. Quando se referem às atividades globais do processo, as regras permitem um controle vertical da integridade e consistência.

4.5. Considerações Finais

Este capítulo estabeleceu as principais exigências impostas a um SBD por sistemas PAC. Em vista destas exigências, definiu-se um modelo de dados, denominado MER/PAC, adequado à modelagem a nível descritivo de aplicações PAC, de modo a ser suportado por um SGBD que supra as necessidades apontadas na seção 4.2. As primitivas de modelagem do MER/PAC são propostas de modo que permitam a modelagem de dois aspectos distintos que caracterizam aplicações PAC, definidos no capítulo 3: o aspecto micro e o aspecto macro de projeto.

O aspecto micro se refere à atividade de projeto em si, onde os dados de um objeto de projeto são estruturados e manipulados de forma a garantir determinadas propriedades definidas para o objeto. Nos estudos e propostas existentes para SGBD's para PAC, este é o aspecto geralmente abordado, não se preocupando, a nível de SGBD, com os possíveis reflexos dos dados de um determinado esquema, referente a um objeto de projeto, sobre os outros esquemas, que representam outros objetos de projeto que coexistem em um sistema PAC.

O aspecto macro caracteriza a sistemática de projeto, o que significa que o próprio processo de projeto é o centro de preocupações. De fato, um processo que age sobre um determinado objeto de projeto (aspecto micro) pode, eventualmente, estar subordinado a um processo de nível superior, que por sua vez, em face dos resultados do subprocesso, pode decidir pela ativação de

outros subprocessos (que agem sobre outros objetos de projeto) ou pela própria mudança da concepção do objeto associado ao subprocesso, o que quer dizer que a nível micro haverá uma mudança no esquema de banco de dados associado ao objeto.

Existem poucos estudos a respeito da aplicação de SGBD's no controle do aspecto macro. A solução proposta pelo MER/PAC para modelar tal aspecto (primitiva esquema) procurou generalizar alguns aspectos abordados em /NEUM 82/, que estabelece um modelo para representação de projeto PAC.

Resumidamente, /NEUM 82/ define seu modelo através das seguintes definições básicas:

A) Representações e Versões: um objeto do mundo real pode ser visualizado sob diversos ângulos e, portanto, um objeto pode ter diferentes bases de dados que o representam. Estes diferentes objetos de banco de dados são denominados Representações. As diferentes instâncias de uma base de dados associada a uma representação denominam-se Versões:

B) Grafo de Derivação e Grafo de Versão: os relacionamentos entre as representações são expressas em /NEUM 82/ na forma de um grafo acíclico dirigido, denominado grafo de derivação. O grafo de derivação especifica as dependências estáticas entre as representações de um objeto de projeto. O grafo de versão é um grafo acíclico dirigido que define as dependências dinâmicas entre as versões individuais, definindo os relacionamentos entre estas, isto é, definindo qual versão é derivada de outra através de uma função de derivação qualquer:

C) **Consistência Global e Local:** As restrições de consistência local são os predicados que dizem respeito às diversas entidades que compõem uma determinada versão. As restrições de consistência global são um conjunto de predicados associados ao conjunto de representações. Assim, diz-se que uma versão VI de uma representação RI é globalmente consistente se e somente se esta versão puder ser derivada a partir do conjunto correspondente de representações de acordo com o grafo de derivação;

D) **Subgrafos e Supergrafos. Inconsistência Dependente e Independente:** O supergrafo de uma versão é o conjunto de todas as versões predecessoras (direta ou indiretamente) no grafo de versão. Analogamente, o subgrafo de uma versão é o conjunto de todas as versões sucessoras no grafo de versão (direta ou indiretamente). Assim, uma versão é dita estar independentemente inconsistente se o supergrafo desta versão é globalmente inconsistente. A partir daí, qualquer versão que pertença ao subgrafo de uma versão independentemente inconsistente é dita ser uma versão dependentemente inconsistente.

Esta visão do aspecto global pode ser representada através das primitivas do MER/PAC, apresentadas na seção 4.4. Utilizando o jargão do MER/PAC, pode-se dizer que o conceito de representação em /NEUM B2/ equivale ao conceito de tipo de esquema no MER/PAC, sendo que um esquema de projeto pode ser modelado também usando a primitiva esquema do MER/PAC. O conceito de versão de /NEUM B2/ tem correspondência direta com o conceito de versão do modelo de dados. Por outro lado, o conceito de expansão de um tipo de esquema no

MER/PAC não tem equivalência no modelo de /NEUM 82/, representando no entanto um aspecto importante no modelo de processo de projeto, pois durante este processo cada esquema está sujeito a alterações não apenas em dados operacionais mas também nas próprias estruturas dos esquemas.

As restrições de consistência local e global têm correspondência direta com as cláusulas de integridade que fazem parte da definição do MER/PAC. No esquema de projeto, tais cláusulas equivalem às restrições de consistência global. No esquema MER/PAC que define cada representação, as cláusulas de integridade representam as restrições de consistência local, restrições estas que são levadas em conta para a definição de uma transação PAC (de acordo com o modelo de /NEUM 82/).

A título de exemplo, será tomado de /NEUM 82/ um conjunto de regras de consistência que devem ser seguidas durante o processo de projeto para garantir a consistência final dos dados:

- Regra 1** uma versão pode ser atualizada se e somente se ela é globalmente consistente ou independentemente inconsistente.
- Regra 2** uma versão somente pode ser criada usando versões globalmente consistentes.
- Regra 3** qualquer versão pode ser removida. Se uma versão é removida então as versões originadas desta devem ser removidas.
- Regra 4** qualquer versão pode ser lida.

A utilização do MER/PAC para a modelagem do aspecto global implica que, após qualquer transação PAC, o SGBD se encarrega de verificar as cláusulas de integridade, conforme definido globalmente, efetuando assim uma análise no sentido vertical na árvore que representa a hierarquia de projeto.

CAPÍTULO 5: EXEMPLO DE APLICAÇÃO

5.1. Introdução

Com o objetivo de mostrar a aplicabilidade do MER/PAC para a estruturação de dados PAC, será apresentado neste capítulo a modelagem de um problema real de projeto a ser ambientado em um sistema PAC. Esta aplicação se refere ao projeto do "lay-out" de placas de circuito impresso.

Uma placa de circuito impresso é um dos vários tipos de implementação física de um circuito eletro-eletrônico. Consta de uma placa de material isolante (fibra de vidro ou equivalente) sendo que em cada face da placa é colocada uma película de metal (geralmente cobre), formando assim uma placa rígida metalizada.

Em seguida, a partir de um gabarito denominado "lay-out" do circuito impresso e que é desenvolvido considerando-se alguma representação genérica do circuito eletro-eletrônico (esquema elétrico ou tabelas de ligações e de componentes), são gerados dados de projeto e de fabricação. A partir destes dados e através de processos químicos e industriais específicos, a película de metal é modificada de forma que o produto final se torne uma placa com trilhas de metal que conectam furos e/ou áreas metalizadas. Nesta placa, os furos e áreas metalizadas correspondem, de modo geral, aos pontos de conexão elétrica dos componentes eletro-eletrônicos, constantes da representação do circuito elétrico. Da mesma forma, as

trilhas de metal correspondem às ligações existentes entre os pontos de conexão destes componentes.

Além desta composição básica, uma placa de circuito impresso conta também com alguns elementos auxiliares. Entre outros, tem-se os elementos de representação gráfica dos componentes eletro-eletrônicos e elementos de referência para a gerência do processo de projeto e de fabricação da placa de circuito impresso.

Os elementos de representação gráfica têm o objetivo de estabelecer uma correspondência biunívoca entre os componentes físicos, que serão eventualmente instalados na placa, e os componentes correspondentes da representação do circuito elétrico (esquema elétrico ou tabela de ligações). Estes elementos gráficos são geralmente "impressos" sobre a placa através de processos específicos ("silk screen" ou equivalente), e são subdivididos em:

- * elementos gráficos - representações simbólicas das diversas classes de componentes eletro-eletrônicos existentes;
- * elementos de texto - identificam o componente físico e estabelecem a correspondência com os componentes da representação do circuito elétrico.

O objetivo do sistema-exemplo, o qual será chamado SPCI (Sistema de Projeto de Circuito Impresso) é o suporte ao projeto de "lay-out" de placas de circuito impresso, a partir de alguma representação do circuito elétrico. Como resultado final o sistema produz toda a informação referente ao "lay-out", necessária ao

processo de fabricação e à gerência do projeto.

O SPCI é constituído de 3 processos básicos:

- * GERÊNCIA: tem por função abrir, acompanhar e finalizar os serviços envolvidos no processo de projeto;
- * MANUTENÇÃO DE PADRÕES: processo responsável pela manutenção dos padrões que são usados nos projetos;
- * EXECUÇÃO DE PROJETO: processo no qual, efetivamente, é executado o projeto da placa de circuito impresso.

A modelagem do SPCI será feita da seguinte forma:

1. Modelagem Conceitual (seção 5.2) : será feita uma descrição geral do problema "projeto do lay-out de um circuito impresso" (correspondente ao processo de EXECUÇÃO DE PROJETO), para em seguida ser feita uma descrição textual detalhada dos objetos e relacionamentos envolvidos no problema;
2. Desenvolvimento do Modelo Descritivo (seção 5.3): utilizando o MER/PAC, será desenvolvido o esquema correspondente aos dados de projeto, a partir do modelo conceitual desenvolvido na fase anterior;
3. Considerações Finais (seção 5.4): utilizar-se-á o MER/PAC na modelagem dos três processos do sistema e seus interrelacionamentos: GERÊNCIA, MANUTENÇÃO DE PADRÕES e EXECUÇÃO DE PROJETO.

5.2. Projeto de Circuito Impresso - Modelagem Conceitual

5.2.1. Descrição Geral

De um modo geral, uma placa de circuito impresso, representada na figura 5.1 pelo seu "lay-out", é composta por uma placa com determinadas características mecânicas e que possui uma área útil, dentro da qual é permitido posicionar componentes e traçar rotas.

O componente é um conjunto de elementos, eletrônicos ou não-eletrônicos, que é identificado de maneira única dentro da placa. Assim, diz-se que um componente é composto por vários componentes, sendo a placa (vazia, sem componentes e rotas) também uma classe de componente.

Um componente contém pontos de acesso elétrico, os pinos, além de uma área de ocupação, geometria e furos de componente. A geometria é uma representação gráfica do componente. Os furos de componente representam os diversos tipos de furos que podem ocorrer em um componente. Correspondem fisicamente aos pontos de interconexão entre os elementos físicos, e/ou aos pontos usados para fixação de um elemento físico na placa.

Os pinos dos componentes são interligados através de rotas, de acordo com uma lista de sinais. As rotas representam as trilhas de cobre que farão parte da placa, após a fabricação. Também fazem

parte da placa os seguintes elementos: máscaras de proibição de furos, máscaras de proibição de rotas, metalizações e furos de sinal, também chamados furos de rota. As máscaras de proibição representam áreas nas quais é proibida a existência de furos ou rotas.

Uma placa de circuito impresso é constituída fisicamente por diversos níveis ("layers") ou camadas, sendo o número destes níveis sempre maior ou igual a 2. O conceito de níveis de uma placa é uma extensão à idéia de "faces de uma moeda", podendo ser dito informalmente que uma placa possui várias "faces". Como há a possibilidade de que uma determinada rota não possa ser localizada totalmente em um único nível, define-se o furo de sinal, que representa o ponto no qual uma rota continua em outro nível da placa.

Neste exemplo são ainda considerados dois conceitos importantes: sinal e ligação. O sinal representa o relacionamento entre os pinos dos diversos componentes de uma placa, sendo identificado por um nome único dado pelo projetista. A ligação é um elemento abstrato que define a conexão entre dois pinos, sendo a rota a realização física de uma ligação.

Para finalizar, assume-se, a princípio, que o projetista somente pode utilizar os componentes, furos e metalizações que estão definidos em uma biblioteca de padrões. A partir desta biblioteca é feita a carga dos dados na estrutura de dados do projeto, sendo que não é permitido ao projetista manipular ou modificar os padrões da

biblioteca.

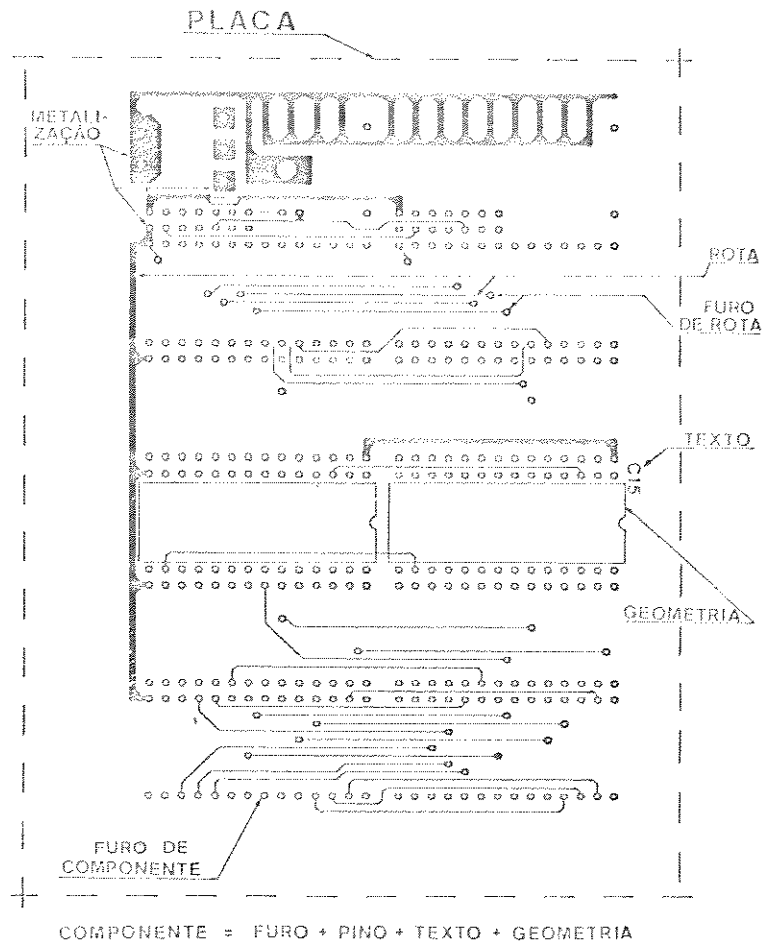


Figura 5.1. Lay-out de um Circuito Impresso

5.2.2. Modelagem Conceitual

Serão agora especificados os objetos da estrutura de dados, os relacionamentos entre estes objetos e restrições válidas nestes relacionamentos. Esta especificação, feita de maneira textual, constituirá o modelo conceitual de projeto, sendo que a descrição formal, o esquema descritivo, será feita nas próximas seções, a

nível de modelo de dados.

5.2.2.1. Grupo A - Regras Referentes a Componentes

1. Para criar um pino, deve-se associá-lo a um componente.
2. Um componente deve ter uma área de ocupação associada.
3. Um componente deve ter um furo de componente associado.
4. Para criar uma máscara de proibição de furo, deve-se associá-la a um componente.
5. Para criar uma máscara de proibição de rota, deve-se associá-la a um componente.
6. Para criar uma máscara de metalização, deve-se associá-la a um componente.
7. Um componente pode ser formado por vários componentes.
8. Um componente deve ter uma geometria e um texto de descrição
9. Um componente que não tem um componente-pai é classificado como placa.
10. Só existe 1 (uma) placa.
11. As posições dos pinos, furos de componente, área de ocupação e máscaras de proibição, em relação ao componente a que estão associados, não podem ser modificadas. Exceção: máscara de proibição se componente é placa.
12. Máscaras de ocupação não podem se sobrepor.

13. Máscaras de ocupação devem sobrepor-se à área útil da placa.
14. Furos e metalizações devem ter uma geometria.

5.2.2.2. Grupo B - Regras Referentes a Rotas

1. Uma rota é definida por uma sequência ordenada de pontos de rota, classificados em:
 1. ponto de geometria (indica mudança na direção da rota)
 2. ponto de furo
 3. ponto de ligação pino-rota
 4. ponto de ligação rota-rota
 5. ponto de pino
 6. ponto de alteração de espessura.

Os pontos extremos desta sequência são os pontos extremos da rota. A inserção é feita entre dois pontos consecutivos, de acordo com a posição dos pontos na placa. Esta regra de inserção não é válida para os dois primeiros pontos inseridos.

2. Uma rota pode pertencer a uma das seguintes classes:
 0. normal
 1. rota de componente não-posicionado
 2. rota de componente posicionado
 3. área de metalização
 4. barramento de douração
 5. cantoneira

6. moldura da máscara de solda.

OBS.: Um componente é dito ser não-posicionado se ele faz parte da composição da placa, mas não se encontra efetivamente posicionado nesta. Caso um componente (posicionado ou não-posicionado) seja retirado definitivamente da placa, diz-se que este componente foi desposicionado.

3. Um furo de sinal pode pertencer a uma das seguintes classes:
 0. normal
 1. furo de componente não posicionado
 2. furo de componente posicionado.
4. Para criar um furo de sinal, deve-se associá-lo a um sinal.
5. Por outro lado, o furo de sinal deve ser associado também a uma rota.
6. Quando se associa um furo de sinal a uma rota R pertencente às classes 0, 1 ou 2, então deve-se associar um ponto de rota a R, cuja classe é 2 (ponto de furo).
7. Para criar um ponto de rota, deve-se associá-lo a uma rota.
8. Uma rota pode ter vários furos de sinal. Um furo de sinal pode ser associado a várias rotas, relativas a um mesmo sinal.
9. Quando se elimina ou se move um furo de sinal de uma rota R, o ponto de rota correspondente se torna um ponto de geometria (ponto de rota classe 1).

10. Quando se cria um furo de sinal associado a uma rota R de tal forma que a posição do furo coincide com a posição de um ponto de geometria de R, então o ponto de rota correspondente a este último se torna um ponto de furo (ponto de rota classe 2).
11. Rotas pertencentes a sinais diferentes não podem se sobrepor (leva-se em conta a espessura da rota).
12. Se um furo de sinal não tem mais nenhuma associação com rotas, então deve ser eliminado.
13. Quando pontos de geometria, que são extremos de rotas, coincidem, tornam-se pontos de ligação nula (ponto de rota classe 4).

No que se refere ao conceito de ponto de rota, pode-se dizer que qualquer ponto de relação (ponto de rota classe 2, 3, 4, 5) é um ponto de geometria de nível mais alto, já que representa também um relacionamento entre objetos. As regras B.10, B.11, C.19 e C.20 representam este conceito.

5.2.2.3. Grupo C - Regras Referentes a Componentes, Rotas, Ligações e Sinais

1. Somente é possível conectar componentes (pinos) através de ligações e rotas.
2. A lista de sinais estabelece uma associação 1:n entre sinal (1) e pinos (n).
3. Um sinal pode pertencer a uma das seguintes classes:
 1. alimentação

2. normal

3. bus.

4. Para criar uma rota, deve-se associá-la a um sinal.
5. Além disso, se a rota pertencer à classe 0 (NORMAL) e se o sinal não for da classe 1 (ALIMENTAÇÃO), ela pode ser criada somente a partir de uma ligação.
6. Para criar uma ligação entre 2 (dois) pinos, estes devem pertencer ao mesmo sinal e os componentes correspondentes aos pinos devem estar posicionados na placa.
7. Para criar uma ligação entre 2 (duas) rotas, estas devem pertencer ao mesmo sinal.
8. Para criar uma ligação entre um pino e uma rota, estes elementos devem pertencer ao mesmo sinal, e o componente correspondente ao pino deve estar posicionado na placa.
9. Seja uma rota R, tal que os pinos V1 e V2 sejam seus pontos de rota (i.e., a rota está resolvida). Então, se o componente correspondente a V1 é desposicionado (retirado da placa), tem-se:
 1. elimina-se a rota R;
 2. segue-se a regra G.10.
10. A eliminação de uma rota R implica na eliminação das ligações e furos de rota associados a R e na eliminação de todos os pontos de rota de R.

11. Seja uma ligação L1 que relaciona dois pinos V1 e V2. Então, se é criada uma rota R associada a L1 e com origem em V1, tem-se:

1. cria-se uma ligação L2 entre V2 e R;
2. o pino V1 se torna ponto de rota de R;
3. o ponto de ligação L2 se torna ponto de rota de R;
4. elimina-se a ligação L1.

12. Seja uma ligação L entre um pino V1 e uma rota R. Então, se a posição do ponto de ligação é igual à posição do pino, a ligação L deve ser eliminada.

OBS.: Um ponto de ligação é um ponto pertencente a uma rota e que é também um dos extremos de uma ligação (rota-rota ou pino-rota).

13. Seja uma rota R que tem os pinos V1 e V2 como pontos de rota. Então, se a posição do componente correspondente ao pino V1 for alterada, tem-se:

1. cria-se uma ligação entre V1 e R;
2. segue-se a regra C.19;
3. elimina-se o pino V1 como ponto de rota de R.

14. Seja uma ligação L1 que está sendo resolvida com uma rota R1, situada no nível F1. Seja ainda a placa composta por n níveis F1, F2, ..., Fn. Então, se é criado um furo de sinal em R1, durante sua resolução, tem-se:

1. cria-se uma rota R2 em um nível F1, diferente de F1;

2. a ligação L1 passa a ser associada a L2 e não mais a R1;
 3. o furo de sinal passa a ser ponto de rota de R1 e R2.
15. Seja um furo de sinal que faz parte das rotas R1 e R2. Então, se o furo vai ser eliminado, tem-se:
1. se R1 e R2 estão em níveis diferentes, o furo de sinal não pode ser eliminado.
 2. se R1 e R2 estão no mesmo layer, o furo é eliminado (furo de sinal e ponto de rota) e cria-se uma ligação nula L1 entre R1 e R2.
16. Seja uma ligação L1 entre duas rotas R1 e R2, que é nula, isto é, os extremos da ligação são coincidentes. Sejam ainda estes pontos de ligação extremidades de ambas as rotas. Então, tem-se:
1. cria-se uma nova rota R3, com as mesmas características (atributos) de R1 e R2;
 2. os conjuntos de pontos de rota de R1 e R2 são unidos e se tornam pontos de rota de R3 (o ponto de ligação nula entre R1 e R2 se torna um ponto de geometria de R3);
 3. quaisquer ligações e/ou furos de sinal que estejam associados a R1 e R2, passam a ser associados a R3.
 4. as rotas R1 e R2 são eliminadas, e segue-se a regra G.10.

17. Seja uma ligação L1 entre um pino V1 e uma rota R1. Na criação de uma rota R a partir de L1, tem-se:
1. Se a origem de R está em V1, então cria-se uma nova rota R2 e uma ligação L2 entre R1 e R2. O pino V1 passa a ser ponto de rota de R2 e ponto de ligação L2 se torna ponto de rota de R1 e de R2. Elimina-se L1, que deixa de existir como ponto de rota de R1;
 2. Se a origem de R está em R1 (no ponto de ligação L1), cria-se um ponto de geometria, associado a R1, com a mesma posição do ponto de ligação L1.
18. Quando se associa uma ligação a uma rota R, então deve-se associar um ponto de rota a R, cuja classe é 3 (ponto de ligação rota-pino) ou 4 (ponto de ligação rota-rota).
19. Quando se elimina uma ligação associada a uma rota R, o ponto de rota correspondente se torna um ponto de geometria (ponto de rota classe 1).
20. Quando se cria uma ligação associada a uma rota R de tal forma que a posição do ligação coincide com a posição de um ponto de geometria de R, então o ponto de rota correspondente a este último se torna um ponto de ligação (ponto de rota classe 3 ou 4).
21. Não é possível criar ligações (pino-rota ou rota-rota) usando rotas pertencentes às classes 3, 4, 5 e 6.

5.2.2.4. Estrutura de Dados e Regras de Manipulação

Nas regras citadas anteriormente, deve-se notar que algumas não podem ser vistas como especificações da estrutura de dados, mas sim como regras de manipulação dos dados. Por regras de manipulação entende-se aquelas que dizem respeito a instâncias específicas dos objetos descritos. Tais regras envolvem as ocorrências (valores) dos objetos descritos e não o relacionamento estrutural entre eles. As regras que dizem respeito à manipulação são, a princípio: A.12, A.13, B.6, B.11, B.12, B.13, B.14, C.5, C.9, C.11, C.13, C.14, C.15, C.16, C.17, C.19 e C.20.

Uma vez estabelecidas as regras de estrutura de dados e as regras de manipulação, pode-se agora estabelecer o Modelo Descritivo através do MER/PAC, o que será feito na próxima seção.

5.3. Modelo Descritivo - MER/PAC

A partir do modelo conceitual estabelecido na seção anterior, serão utilizadas as primitivas MER/PAC para estabelecer o modelo descritivo ou esquema do projeto de "lay-out" de um circuito impresso. As figuras 5.2 a 5.7 apresentam os diagramas MER/PAC correspondentes a este esquema.

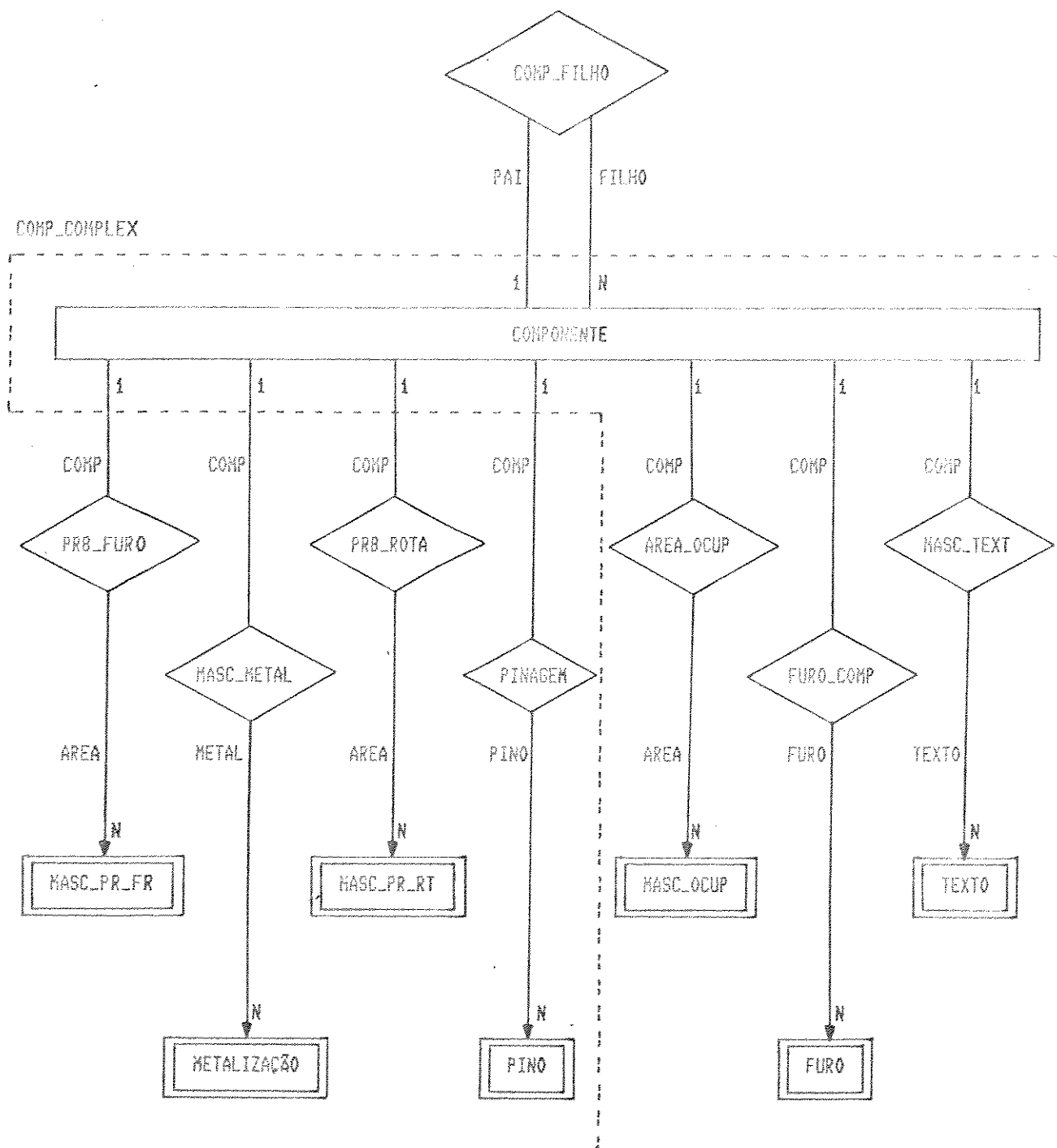


Figura 5.2. Diagrama MER/PAC - Esquema de Projeto - Grupo A

Na figura 5.2, procura-se modelar um componente como uma entidade complexa, que é constituído pelos tipos de entidades fracas TEXTO, FURO e MASC_OCUP (máscara de ocupação). Isto quer dizer que,

ao se criar um componente, devem ser criadas entidades texto, furo e máscara de ocupação. Por outro lado, como um componente não precisa ter necessariamente máscaras de proibição de furo (entidade MASC_PR_FR e relacionamento PRB_FURO), máscaras de proibição de rota (entidade MASC_PR_RT e relacionamento PRB_ROTA), metalizações e pinos, os tipos de entidade associados são fracos, mas não subordinados a entidade complexa. Assim, a entidade complexa é modelada através do agrupamento COMP_COMPLEX, que possui regras de formação que seguem a especificação citada anteriormente.

O relacionamento COMP_FILHO modela diretamente a regra A.7 do modelo conceitual. A figura 5.3 modela o fato de que componentes, furos e metalizações devem ter uma geometria, existindo aí uma imposição de existência. As regras de formação do agrupamento ELEM_GEOM são especificadas de forma a garantir não só a imposição de existência, como também uma ocorrência mutuamente exclusiva dos tipos de relacionamentos FURO_GEOM, METAL_GEOM, COMP_GEOM, pois uma geometria associada a uma metalização não pode ser a mesma associada a um furo de componente ou a um componente, e assim por diante. Observa-se ainda na figura 5.3 que os tipos de entidade METALIZAÇÃO e FURO são fracos, mas não em relação ao tipo de entidade GEOMETRIA, posto que os relacionamentos que associam estas entidades não estabelecem a dependência de existência, embora também sejam relacionamentos fracos.

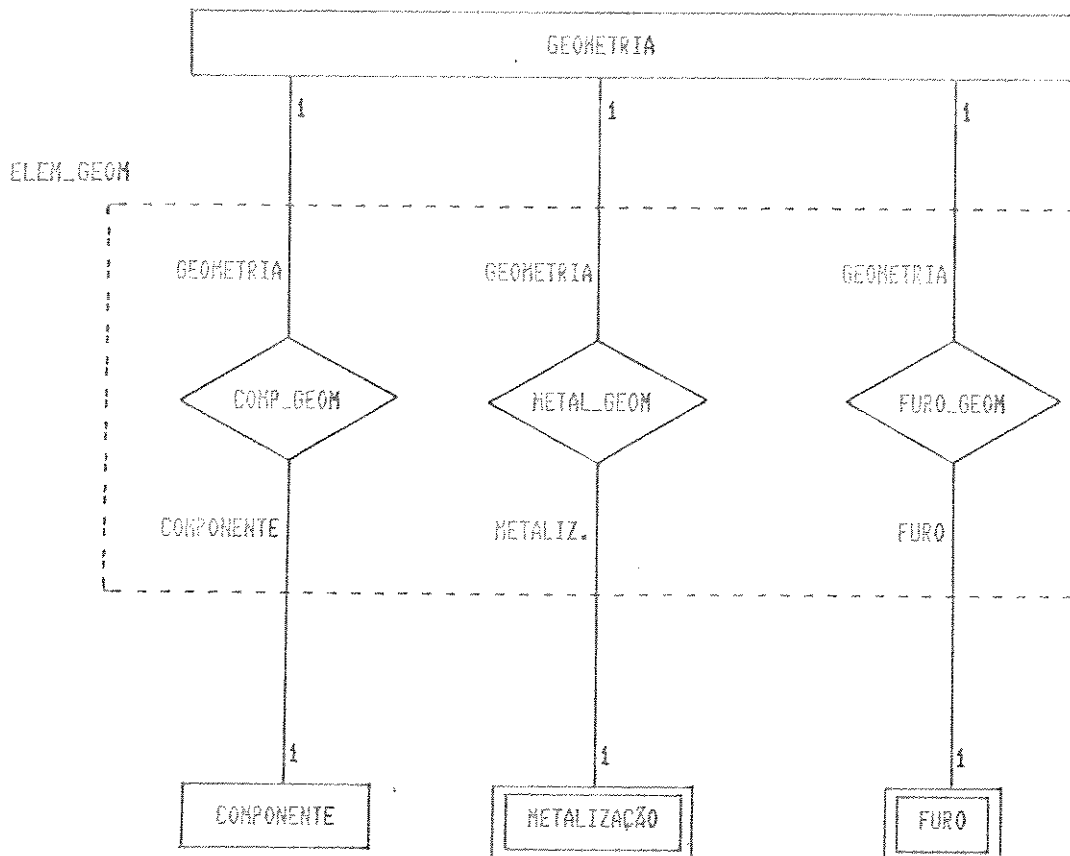


Figura 5.3. Diagrama MER/PAC - Esquema de Projeto - Grupo A

A figura 5.4 dá um exemplo de definição de atributos para algumas das entidades definidas na figura 5.2, e que foram omitidos do modelo conceitual apenas para efeito de simplificação. Observe-se que o atributo REGIÃO de MASC_PR_FR (máscara de proibição de furo) é associado ao domínio de valores RETÂNGULO que, a nível organizacional, deve ser definido através de um TAD retângulo. Outro elemento a ser notado é a multivaloração de atributos. Na figura, os atributos ATIVAÇÃO e NOME, do tipo de entidade PINO, indicam em quais faces da placa de circuito a conexão elétrica é válida, sendo que a multivaloração é indicada pelos símbolos "1" e "N" na definição do referido atributo.

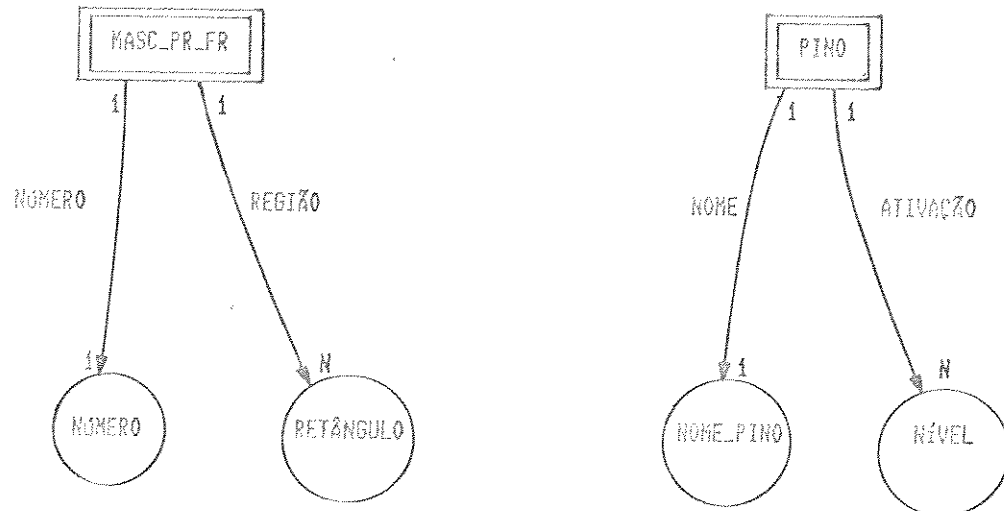


Figura 5.4. Diagrama MER/PAC - Definição de Atributos

As figuras 5.5 e 5.6 apresentam a parte do esquema correspondente às regras de definição de rotas (seção 5.2.2.2). Note-se que o tipo de entidade FURO_SINAL é fraco em relação aos tipos de entidades SINAL e ROTA, sendo que tais dependências modelam as regras B.4 e B.5. Na figura 5.6, o agrupamento ELEM_PONTO (definido, através de regras de formação, como três tipos de relacionamentos mutuamente exclusivos) representa o fato de que um ponto de rota pode ser também um furo de componente, um furo de sinal ou um pino. Observa-se ainda na figura 5.6 que os tipos de entidade FURO_SINAL e FURO são fracos, mas não em relação ao tipo de entidade PONTO_ROTAS, posto que os relacionamentos que associam estas entidades não estabelecem a dependência de existência, embora também sejam relacionamentos fracos.

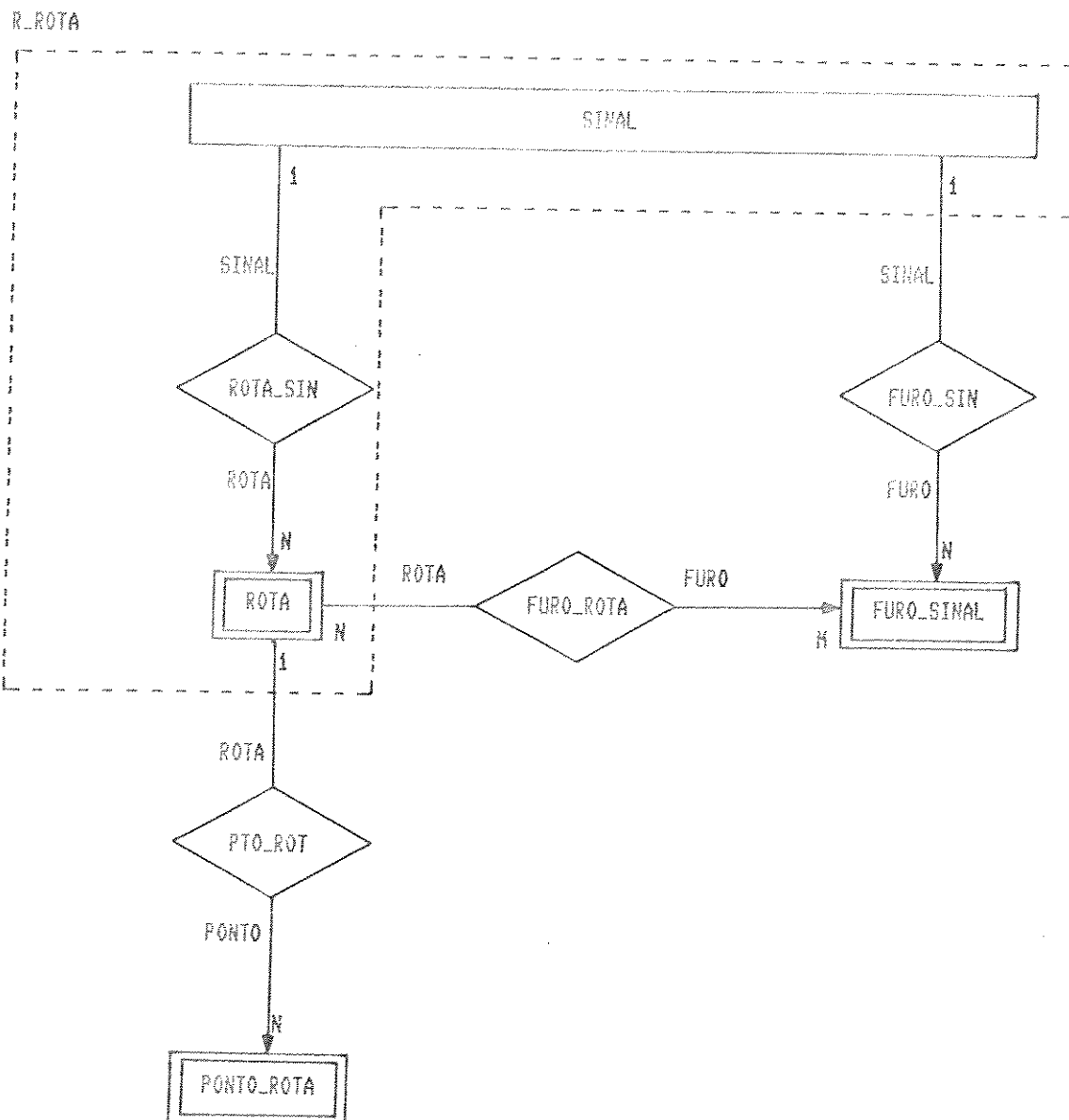


Figura 5.5. Diagrama MER/PAC - Esquema de Projeto - Grupo B

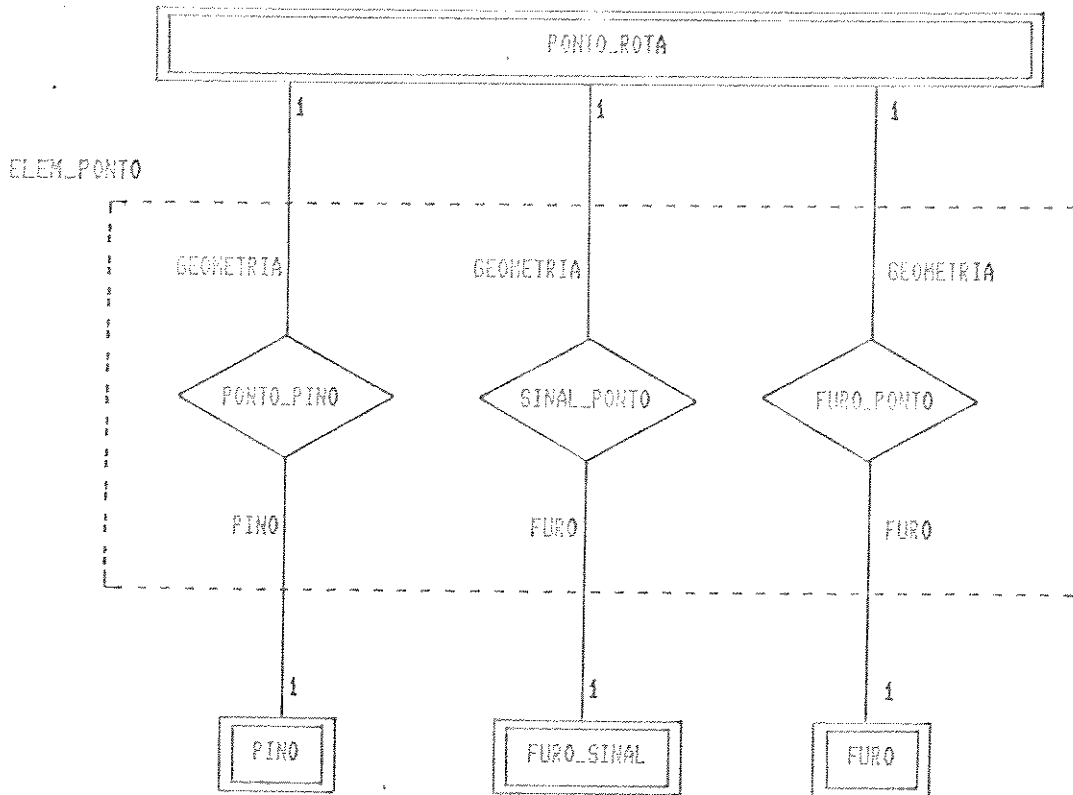


Figura 5.6. Diagrama MER/PAC - Esquema de Projeto - Grupo B

A figura 5.7 se refere às regras que envolvem a interação entre componentes, rotas, ligações e sinais (seção 5.2.2.3). As ligações são modeladas através de relacionamentos que envolvem os agrupamentos R_ROTA (figura 5.5) e P_PINO (figura 5.7). Desta forma, as ligações envolvem apenas rotas e pinos que possuem sinais associados. As regras de integridade se encarregam de verificar se os elementos a serem envolvidos em uma ligação possuem o mesmo sinal.

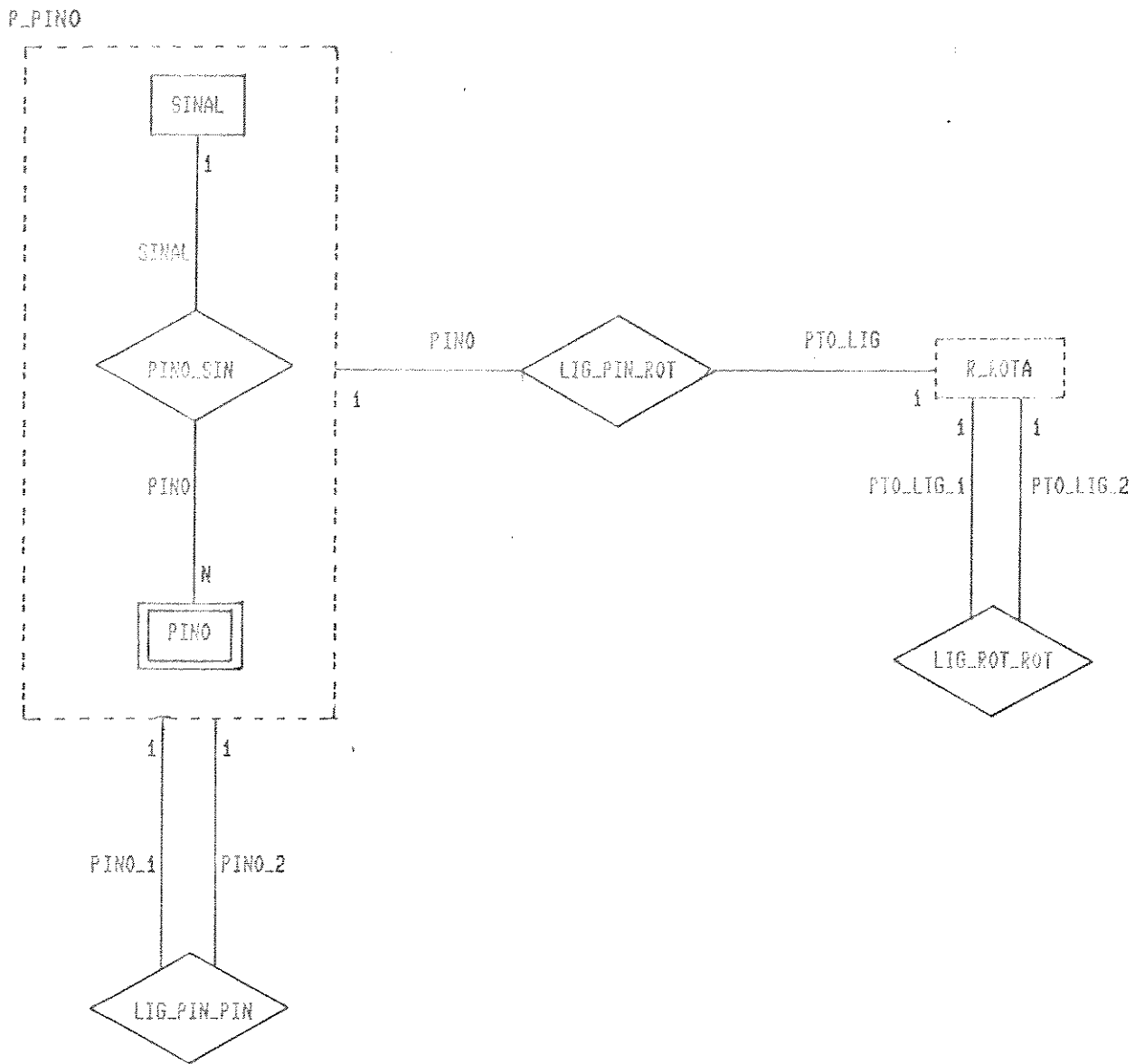


Figura 5.7. Diagrama MER/PAC - Esquema de Projeto - Grupo C

5.4. Modelagem do Processo de Projeto

A modelagem desenvolvida até aqui está relacionada com o processo EXECUÇÃO DE PROJETO. O próximo passo é modelar as informações levando em consideração a existência dos outros processos existentes no SPCI: GERÊNCIA e MANUTENÇÃO DE PADRÕES.

Como foi definido no modelo conceitual (seção 5.2.1), só é possível usar componentes que já estejam definidos em uma Biblioteca de Padrões, que contém elementos pré-definidos pelo processo de MANUTENÇÃO DE PADRÕES. Assim, só é permitido ao processo de EXECUÇÃO DE PROJETO evoluir se existir determinado conjunto de padrões de elementos que possa ser usado.

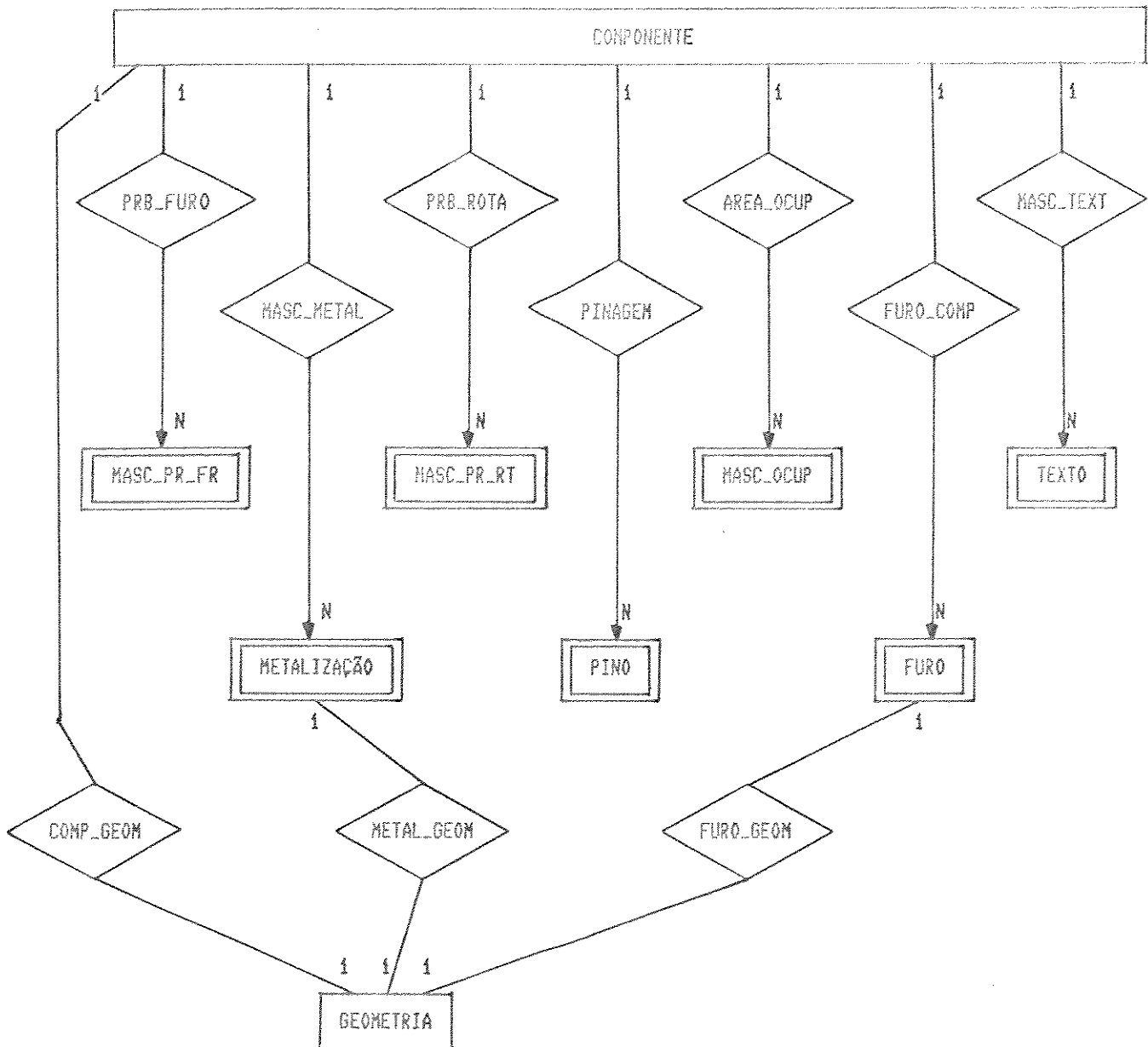


Figura 5.8. Esquema para Biblioteca de Padrões

Por outro lado, a construção de padrões envolve um modelo conceitual (não definido aqui) que é ligeiramente diferente do modelo associado ao processo EXECUÇÃO DE PROJETO, descrito na seção 5.2. Por exemplo, para se definir um componente padrão, não são necessárias ligações e sinais, sendo que os pinos são ligados diretamente através de rotas, sem quaisquer restrições. Como consequência, tem-se um modelo descritivo diferente, isto é, um esquema diferente para ser usado pelo processo de MANUTENÇÃO DE PADRÕES. Um exemplo deste esquema pode ser visto na figura 5.8, através de um diagrama MER/PAC.

Para representar, a nível de estrutura de dados, o interrelacionamento entre os esquemas definidos para os processos EXECUÇÃO DE PROJETO (EP) e MANUTENÇÃO DE PADRÕES (MP), é utilizada a primitiva esquema do MER/PAC (figura 5.9). O tipo de esquema BIBLIOTECA representa a estrutura de dados utilizada pelo processo MP, sendo que o esquema da figura 5.8 representa uma ocorrência (expansão) para este tipo de esquema. O esquema desenvolvido na seção anterior corresponde, por sua vez, a uma ocorrência (expansão) do tipo de esquema PLACA_GI, e é sobre esta primitiva que o processo EP age.

Deve-se observar os atributos definidos para ambos os tipos de esquema, representam atributos inerentes a primitiva. O atributo versão identifica um conjunto de dados na base de dados, que por sua vez está ligado ao esquema identificado pelo atributo esquema_id.

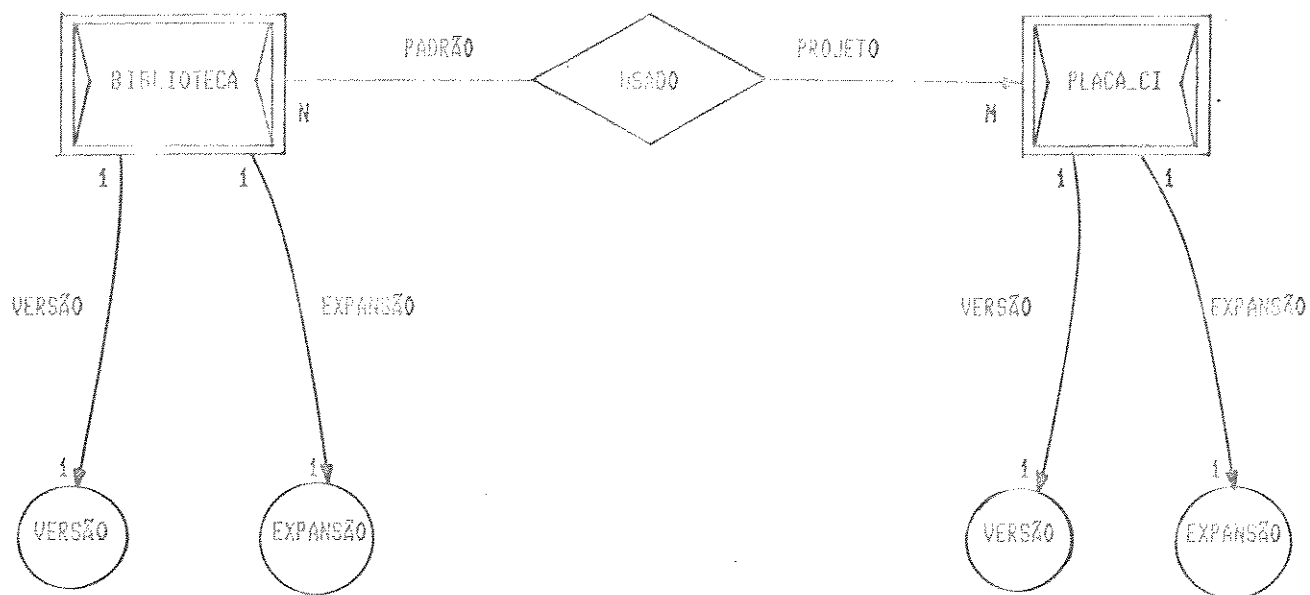


Figura 5.9. Primitivas Esquema associadas ao Projeto de CI

O relacionamento USADO estabelece, a nível de esquemas, o interrelacionamento entre os processos MP e EP. Note-se que o tipo de esquema PLACA_CI é fraco em relação a BIBLIOTECA. Isto quer dizer que, só é possível desenvolver os dados de um esquema PLACA_CI se já existir uma biblioteca de padrões, isto é, um esquema BIBLIOTECA e versões associadas.

Para gerenciar os processos MP e EP, surge o processo GERÊNCIA (GR), que se preocupa com a ativação e acompanhamento destes processos. Tal interrelacionamento entre os processos é mostrado na figura 5.10, onde são representados os processos acima citados, de uma maneira esquemática.

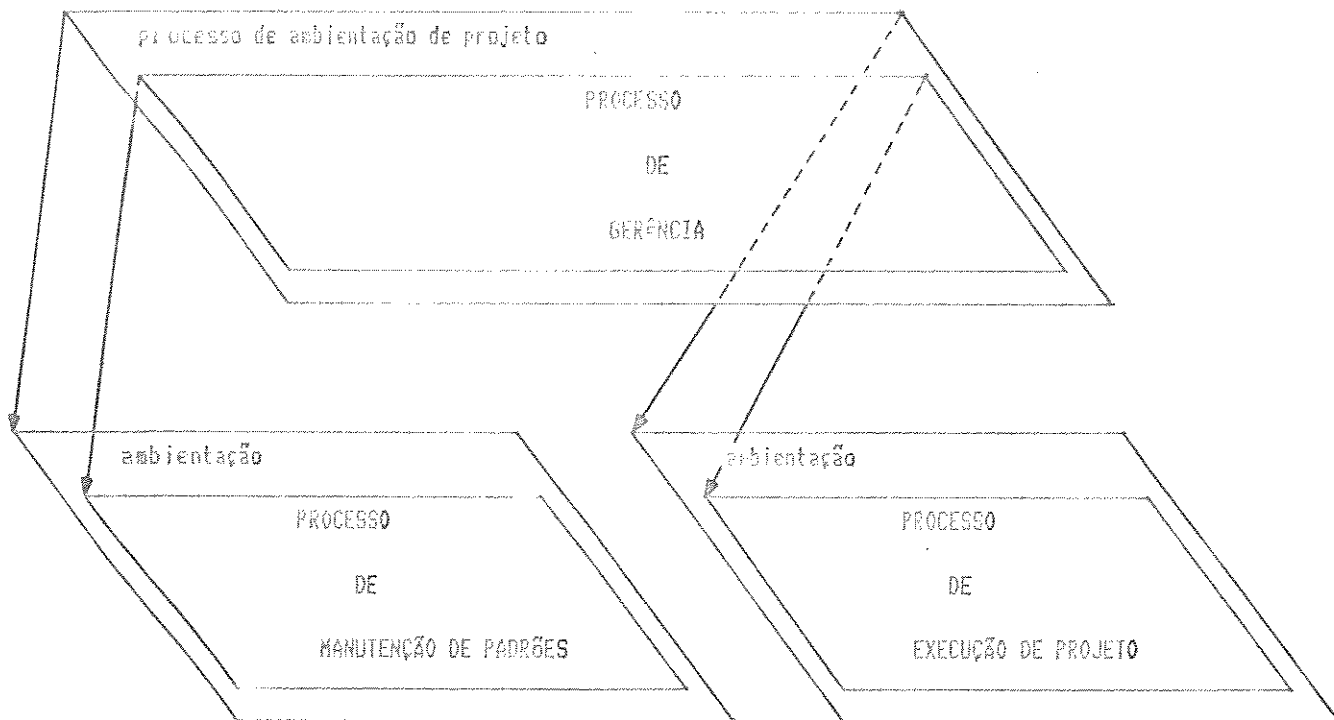


Figura 5.10. Hierarquia dos Processos SPCI

A figura 5.11 mostra o interrelacionamento dos três processos a nível de esquemas. Para cada processo MP e EP existe um conjunto esquema-versão para PLACA_CI e BIBLIOTECA, sendo que a cada um destes conjuntos está associado um par esquema-versão de SUPERVISÃO, que contém o esquema e informações operacionais pertinentes ao processo GR. A estrutura definida na figura 5.11 é, na verdade, um esquema do tipo de esquema SPCI, que representa assim, toda a estrutura de dados associada à atividade de projeto de "lay-out" de um circuito impresso.

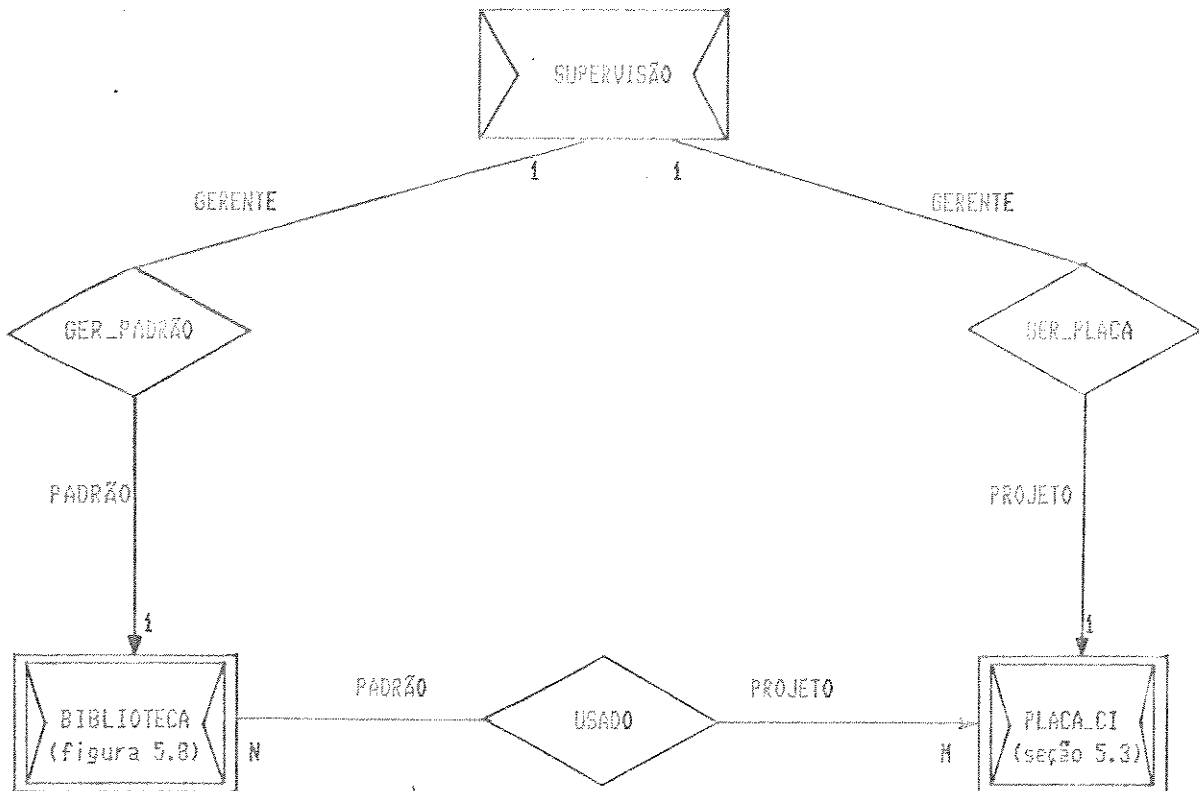


Figura 5.11. Expansão do Esquema SPCI

Deve-se observar, assim, que a utilização da primitiva esquema estabelece, "a priori", qual tipo de ação pode ser tomada pelos processos existentes no ambiente PAC sobre os esquemas existentes. Tais ações são tomadas em função das regras de consistência definidas em cada esquema. Por exemplo, as regras de consistência definidas para um esquema BIBLIOTECA são analisadas pelo processo MP, sendo que as regras definidas para um esquema SUPERVISÃO e/ou SPCI são analisadas pelo processo GR.

O resultado final é que com as mesmas primitivas MER/PAC foi possível modelar tanto os aspectos micro da aplicação como os aspectos macro. Os esquemas definidos na seção 5.3 e figura 5.8

representam o aspecto micro do processo de projeto, e as estruturas definidas nas figuras 5.9 e 5.11 representam o seu aspecto macro.

CAPÍTULO 6. CONCLUSÕES

Atualmente, tem sido dada grande ênfase ao uso de sistemas automatizados para o controle de processos de produção, tendo como objetivos a melhoria da qualidade dos produtos, o aumento da produtividade e a conseqüente diminuição de custos de produção. A evolução dos sistemas computacionais teve um grande impacto na evolução de processos automatizados, dando origem a novos ramos de atividade, decorrentes da integração entre ciência de computadores e engenharia, dentre as quais se destaca o ramo de PAC - Projeto Auxiliado por Computador.

O principal problema de se ambientar processos de projeto a sistemas computacionais está ligado ao controle das informações necessárias ao desenvolvimento de um projeto (documentação, desenhos, dados técnicos de projeto, etc.) e das informações que fluem entre os diversos processos de projeto que constituem o sistema PAC. Tal complexidade de gerência de dados vem tornando mandatário o uso de Sistemas de Gerência de Banco de Dados em sistemas PAC. Este fato motivou o presente trabalho, que procura assim estudar as características de PAC e quais seus reflexos sobre as características de SGBD's para PAC e, portanto, sobre Modelos de Dados voltados para PAC.

Primeiramente procurou-se estudar as características gerais de Sistemas de Gerência de Banco de Dados (capítulo 2), dando-se ênfase aos NÍVEIS DE ABSTRAÇÃO DE DADOS que podem ser suportados por

tais sistemas. A partir daí, fez-se um estudo de três Modelos de Dados - Rede, Relacional e Entidade-Relacionamento - estabelecendo-se suas principais características em termos de poder de modelagem. A partir daí, escolheu-se o MER como Modelo de Dados-base, para uso no desenvolvimento posterior do trabalho.

A seguir, os sistemas PAC foram estudados e caracterizados (capítulo 3) sob o ponto de vista funcional e sob o ponto de vista de modelagem das informações. Neste ponto procurou-se caracterizar as informações PAC sob dois aspectos: aspectos micro, referentes às ações sobre um objeto de projeto; e aspectos macro, referentes à sistemática de projeto.

Após estes estudos, estabeleceu-se um Modelo de Dados adequado à representação dos aspectos micro e macro de PAC, levando-se em conta as considerações sobre Modelos de Dados do capítulo 2. O Modelo de Dados proposto nesta fase do trabalho - Modelo Entidade-Relacionamento para PAC (MER/PAC) permite uma formalização da concepção dos usuários de Aplicações PAC, conforme exemplificado no Capítulo 5 (projeto de placas de circuito impresso).

O MER/PAC é um modelo que se situa no Nível de Abstração Descritivo, visto que através deste modelo é possível representar formalmente os conceitos abstraídos do Mundo Real, sem entrar nos detalhes organizacionais (chaves, índices, registros, etc.).

Com isso, mostrou-se a viabilidade da utilização de

Sistemas de Banco de Dados para tratamento de informações em Sistemas PAC, visto que um dos requisitos básicos deste tipo de aplicação é a transmissão precisa ao Sistema de Banco de Dados do modelo dos objetos de projeto que coexistem no sistema PAC, dos seus interrelacionamentos e dos processos de projeto que agem sobre eles. Desta forma, o ABD (Administrador de Banco de Dados) e/ou o SGBD (Sistema de Gerência de Banco de Dados) podem estruturar adequadamente os dados considerados relevantes aos objetivos da Aplicação. A vantagem de se utilizar um SGBD que suporte um modelo de dados descritivo (e.g. MER/PAC) está no fato de que não há necessidade de mapeamento do modelo descritivo para um modelo organizacional (e.g. relacional ou rede), que seria então submetido ao SGBD. Tal mapeamento é feito diretamente pelo SGBD-MER a partir do modelo descritivo.

Sobre o Modelo proposto, uma extensão adaptada do Modelo Entidade-Relacionamento (MER) proposto em /CHEN 76/, podem-se apresentar as seguintes conclusões:

1) Eficiência do Modelo para descrição de aplicações PAC:

São introduzidas no MER/PAC duas primitivas intimamente relacionadas com aplicações PAC. A primitiva AGRUPAMENTO permite a descrição de objetos e relacionamentos complexos, bem como a agregação e generalização de primitivas. A primitiva ESQUEMA permite a modelagem do aspecto macro de PAC, na medida em que se consegue representar o interrelacionamento entre diversos esquemas, sobre os quais atuam diferentes processos de

projeto.

O grau de modelagem semântica no MER/PAC é maior, que nos outros modelos citados neste trabalho (relacional e rede), o que decorre diretamente do fato de se ter adotado a abordagem MER. A estruturação dos dados em Entidades e Relacionamentos, ao invés de Relações (modelo relacional) ou Registros e Conjuntos (modelo rede), é mais adequada à modelagem PAC em particular, pois permite uma especificação mais clara da natureza dos elementos de projeto a serem estruturados.

O MER/PAC apresenta uma vantagem importante em relação aos outros modelos de dados, pois engloba em suas primitivas conceitos importantes como Agregação, Generalização, Exclusão Mútua e Objetos Complexos. Embora trabalhos recentes tentem englobar tais conceitos nos modelos relacional e entidade-relacionamento, a sua implementação e seu grau de clareza semântica deixam a desejar.

2) Eficiência do Modelo sob o ponto de vista de integridade e consistência:

Com a utilização de regras e com os conceitos de dependência de existência e imposição de existência, o MER/PAC permite a definição clara das restrições de integridade e consistência, bem como das regras de formação de ocorrências de primitivas.

Na verdade, uma vez que o MER/PAC é baseado em tipos de entidades e tipos de relacionamentos, com atributos que fazem um mapeamento destes elementos para

domínios de valores, pode-se propor, a posteriori, que regras sejam na verdade um atributo multivalorado, cujo domínio de valores estabelece a estrutura sintática dos valores (na verdade, as regras) deste atributo. Desta forma, A verificação de regras consiste basicamente na leitura dos valores de um atributo multivalorado e do posterior tratamento destes valores, por exemplo, por um Sistema de tratamento de regras vinculado a um SGBD que suporte o MER/PAC.

3) Eficiência do Modelo no que se refere à uniformidade de descrição dos aspectos micro e macro de PAC:

Através do MER/PAC consegue-se expressar, através de uma única ferramenta de modelagem, tanto os aspectos micro de PAC (definição estrutural dos objetos de projeto), quanto aos aspectos macro de PAC (definição estrutural do ambiente de projeto).

Esta é a principal contribuição do MER/PAC na área de modelagem de dados. Os atuais estudos de banco de dados para PAC pouca ou nenhuma importância têm dado ao aspecto de modelagem do ambiente de projeto através de um modelo de dados. Isto porque nenhum dos modelos de dados existentes fornece ferramentas para tanto, sendo utilizados apenas na modelagem micro. Ao se modelar ambientes de projeto através da primitiva ESQUEMA, determinados ambientes que compõem um ambiente maior podem ser tratados e modelados de maneira análoga a entidades.

O trabalho procurou se deter na definição das primitivas de modelagem do MER/PAC. Não se alcançou neste trabalho o detalhamento de um subtema de importância: a especificação das regras, que são importantes na manutenção da integridade e consistência dos dados. Tal problema constitui tema de um trabalho que pretende dar seqüência ao atual, a nível de curso de doutoramento.

O desenvolvimento deste trabalho se insere em um contexto mais amplo, que consiste no projeto e implementação de um SGBD que suporte o MER/PAC, e que possa ser utilizado em aplicações PAC. Este sistema será, a princípio, composto de três módulos:

1. LIGG: é um programa que pretende servir como interface entre o usuário e o SGBD, através de uma Linguagem Gráfica para Grafos de entidades e relacionamentos (/QUEZ 86/);

2. GERPAC: é o núcleo do SGBD proposto, denominado Sistema de Gerência para PAC, e que manipula as entidades, relacionamentos, esquemas e agrupamentos como definidos pelo MER/PAC (/RICA 86/);

3. SIGA: é o núcleo de gerência da base de dados, denominado Sistema de Gerência Associativa. Este núcleo servirá de interface entre o GERPAC e a base de dados (/RICA 86/).

Finalmente, são sugeridos como desenvolvimentos futuros dentro da linha de pesquisa deste trabalho, os seguintes temas:

1) desenvolvimento e aperfeiçoamento do modelo MER/PAC para descrição de Aplicações PAC;

2) estudo detalhado do aspecto macro de PAC, dando especial atenção ao controle de transações PAC, versões e esquemas (abordagem /NEUM 82/ e MER/PAC);

- 3) extensão da LIGG mencionada anteriormente, visando uma especificação completa para a Linguagem de Definição de Dados e para a Linguagem de Manipulação de Dados, dando-se especial atenção à especificação de regras.
- 4) estudo, proposta e implementação de um sistema de tratamento de regras, a ser utilizado pelo SGBD (GERPAC) na verificação das regras de integridade, consistência e de formação.

CAPÍTULO 7. BIBLIOGRAFIA

- /BERZ 83/ Berztiss, A.T. & Thatte, S.: "Specification and Implementation of Abstract Data Types", Advances in Computers, vol.22, pp. 295-353, Academic Press, 1983.
- /CHAL 82/ Chailis, M.F.: "Typing in Data Base Models", in /ENCA 82/, pp. 265-279.
- /CARV 82/ Carvalho, M.A. & Golendziner, L.G.: "Um Sistema Operacional para Suporte a Banco de Dados", IV Congresso Nacional de Informática SUCEBU, Rio de Janeiro, Outubro, 1982.
- /CHEN 76/ Chen, P.P.-S.: "The Entity-Relationship Model - Toward a Unified View of Data", ACM Transactions on Database Systems, vol.1, no.1, March 1976, pp. 9-36.
- /CHEN 77/ Chen, P.P.-S.: "The Entity-Relationship Approach to Logical Data Base Design", The Q.E.D. Monograph Series - Data Base Management, no.6, 1977.
- /CHEN 80/ Chen, P.P.-S. (Ed.): "Entity-Relationship Approach to Systems Analysis and Design", Proceedings of the International Conference on Entity-Relationship Approach to Systems Analysis and Design, Los Angeles, December, 1979. North-Holland Publishing Company, 1980.
- /CHEN 81/ Chen, P.P.-S. (Ed.): "Entity-Relationship Approach to Information Modeling and Analysis", Proceedings of the 2nd. International Conference on Entity-Relationship Approach, Washington, D.C., October 1981.
- /CODD 70/ Codd, E.F.: "Relational Model of Data for Large Shared Data Banks", Communication of the ACM, Vol.13, no.6, June 1970, pp. 377-397.
- /CODD 72/ Codd, E.F.: "Further Normalization of the Data Base Relational Model", in Rustin, R.: "Data Base Systems - Courant Computer Science Symposium 6", pp. 33-64, Prentice-Hall, Englewood Cliffs, New Jersey, 1972.
- /CODD 74/ Codd, E.F.: "Recent Investigations in Relational Data Base Systems", Proceedings of IFIP Congress 1974, pp. 1017-1021, North-Holland, Amsterdam.
- /DATE 83/ Date, G.J.: "An Introduction to Database Systems: volume II", Addison-Wesley Publishing Company, 1983.
- /DATE 84/ Date, G.J.: "Introdução a Sistemas de Banco de Dados", Editora Campus, Rio de Janeiro, 1984.

- /DAVIS 83/ Davis, C.G.; Jajodia, S.; NG, P.A.-B. & Yeh, R.T. (Eds.): "Entity-Relationship Approach to Software Engineering", Proceedings of the 3rd. International Conference on Entity-Relationship Approach, Anaheim, October, 1983. North-Holland Publishing Company, 1983.
- /DELGA 85/ "Capturing the Semantics of CAD through a Data Model", A.L.N. Delgado & L.P. Magalhães, Proceedings of the 2nd IASTED International Conference Telecommunication and Control - TELECON'85, Dez 10-13, Rio de Janeiro, 1985.
- /DELGA 86a/ "Sistemas de Gerenciamento de Banco de Dados para P.A.C.", A.L.N. Delgado, L.P. Magalhães & I.L.M. Ricarte, Anais do I Simpósio Brasileiro de Banco de Dados - SBBD, Abril 1986, Rio de Janeiro.
- /DELGA 86b/ "Assimilando a Semântica de P.A.C. através de um Modelo de Dados", A.L.N. Delgado & L.P. Magalhães, VI Congresso da Sociedade Brasileira de Computação - XIII Seminário Integrado de Software e Hardware, 19 a 25 de Julho 1986, Recife.
- /EAST 82/ Eastman, C.M. e Lafue, G.M.E.: "Semantic Integrity Transactions in Design Databases". In /ENCA 82/, pp. 45 - 57.
- /EBER 82/ Eberlein, W. & Wedekind, H.: "A Methodology for Embedding Design Databases into Integrated Engineering Systems", in /ENCA 82/, pp. 3-43.
- /ENCA 82/ Encarnação, J. & Krause, F.-L. (Eds.): "File Structures and Data Bases for CAD", North-Holland Publishing Company, Amsterdam, 1982.
- /ENCA 83/ Encarnação, J. e Schlechtendahl, E.G.: "Computer-Aided Design - Fundamentals and System Architecture", Springer-Verlag, 1983.
- /FAGIN 77/ Fagin, R.: "Multivalued Dependencies and a New Normal Form for Relational Database Systems", ACM Transactions on Database Systems, Vol.2, no.3, September, 1977.
- /FAGIN 79/ Fagin, R.: "Normal Forms and Relational Database Operators", Proc. 1979 ACM SIGMOD International Conference on Management of Data.
- /FINK 83/ Finkelstein, L. & Finkelstein, A.C.W.: "Review of Design Methodology", IEE Proceedings, vol.130, part A, no. 4, Junho 1983, pp. 213 - 222.
- /FURT 79/ Furtado, A.L. & Santos, C.S.: "Organização de Bancos de Dados", Editora Campus, Rio de Janeiro, 1979.

- /GRABO 82/ Grabowski, H. e Elgner, M.: "A Data Model for a Design Data Base". In /ENCA 82/, pp. 117 - 151.
- /HAMM 76/ Hammer, M.: "Data Abstractions for Data Base", Bulletin of ACM SIGMOD, Vol.8, no.2, 1976.
- /HARTELT 86/ Hartelt, H.I.M.: "GAV - Um Gerenciador de Áreas de Visualização Baseado na Norma GKS", Tese de Mestrado apresentada a Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas, Dezembro 1986.
- /HATVA 77/ Hatvany, J.: "The Distribution of Functions In Manufacturing Systems". In D. McPherson (Ed.) "Advances In Computer-Aided Manufacture", North-Holland Publishing Company, Amsterdam, 1977, pp. 23-28.
- /ISO 82/ Draft International Standard ISO/DIS 7842. Information Processing. Graphical Kernel System (GKS) - Functional Description. November 1982.
- /ISO 85/ ISO/TC97/SC21/N599. Information Processing Systems. Computer Graphics. Functional Specification of the Programmers' Hierarchical Interactive Graphics Standard (PHIGS). PHIGS Working Draft. April 1985.
- /ISO 86/ ISO/TC97/SC21/N1179. Information Processing Systems. Computer Graphics. Interfacing Techniques for Dialogues with Graphical Devices. CGI Working Draft. May 1986.
- /JOHNS 83/ Johnson, H.R.; Schweitzer, J.E. e Warkentine, E.R.: "A DBMS Facility for Handling Structured Engineering Entities", IEEE Proceedings, 1983, pp. 3 - 11.
- /KAHN 80/ Kahn, B.K.: "LDDM - A Structured Logical Database Design Methodology", Tutorial on Software Design Techniques, 3rd. Edition, 1980.
- /KNUT 73/ Knuth, D.E.: "The Art of Computer Programming - Sorting and Searching", Addison-Wesley Publishing, Massachusetts, 1973.
- /LORIE 82/ Lorie, R.A.: "Issues in Databases for Design Applications". In /ENCA 82/, pp. 213 - 229.
- /MART 77/ Martin, J.: "Computer Database Organization", Prentice-Hall, 1977.
- /NEUM 82/ Neumann, T. e Hornung, G.: "Consistency and Transactions in CAD Database", Proceedings of the Eighth International Conference on Very Large Data Bases - Mexico City, September 1982.

- /NEWM 83/ Newman, W.M. & Sproull, R.F.: "Principles of Interactive Computer Graphics", McGraw-Hill, New York, 1983.
- /NG 81/ Ng, P.A.: "Further Analysis of the Entity-Relationship Approach to Database Design", IEEE Transactions on Software Engineering, Vol SE-7, no.1, 1981.
- /QUEZ 86/ Quezada G., S.: "LIGG-MER/PAG - Uma Linguagem Gráfica Baseada em Grafo para o MER/PAG", versão preliminar de Tese de Mestrado, Faculdade de Engenharia Elétrica, UNICAMP, Dezembro 1986.
- /RICA 85/ Ricarte, J.L.M.: "Definição e Implementação de um Modelo de Dados para o Núcleo CORAS-UNICAMP", 30. Relatório de Atividades - FAPESP, Agosto de 1986.
- /SETZ 86/ Setzer, V.W.: "Projeto Lógico e Projeto Físico de Banco de Dados", V Escola de Computação, Belo Horizonte, 1986.
- /SILVA 85/ Silva, M.V.: "Processador de Diálogo - Uma Ferramenta para Geração de Sistemas Interativos", Tese de Mestrado apresentada a Faculdade de Engenharia de Campinas da Universidade Estadual de Campinas, Abril 1985.
- /SMIT 77a/ Smith, J.M. & Smith, D.C.P.: "Database Abstractions: Aggregation and Generalization", ACM Transactions on Database Systems, vol.2, no.2, June 1977, pp. 105-133.
- /SMIT 77b/ Smith, J.M. & Smith, D.C.P.: "Database Abstractions: Aggregation", Communications of the ACM, vol.20, no.6, June 1977, pp. 405-413.
- /SMIT 80/ Smith, D.C.P. & Smith, J.M.: "Conceptual Database Design", Tutorial on Software Design Techniques, 3rd Edition, 1980.
- /STON 83/ Stonebraker, M.; Rubenstein, B. & Guttman, A.: "Application of Abstract Data Types and Abstract Indices to CAD Data Bases", Proceedings on Engineering Design Applications of ACM-IEEE Database Week, May 1983, pp. 107-113.
- /TAYL 76/ Taylor, R.W. & Frank, R.L.: "Codasy1 - Database Management Systems", Computing Surveys, Vol.8, no.1, March, 1976.
- /TING 83/ Ting, W.S.-: "Implementação de um Núcleo de Banco de Dados baseado na filosofia de CORAS - Documentação Preliminar", relatório interno do Setor de Automação da Faculdade de Engenharia da UNICAMP, 1983.

- /TING 84/ Ting, W.S.: "MERB: Um Modelo de Dados para Aplicação em Controle de Processo", Tese de Mestrado apresentada a Faculdade de Engenharia de Campinas da Universidade Estadual de Campinas, Setembro 1984.
- /TSIC 77/ Tsihritzis, D.C. & Lochovsky, F.H.: "Data Base Management Systems", Academic Press, New York, 1977.
- /TSIC 82/ Tsihritzis, D.C. & Lochovsky, F.H.: "Data Models", Prentice-Hall, Inc., Toronto, 1982.
- /ULLM 80/ Ullman, J.D.: "Principles of Database Systems", Computer Science Press, 1980.
- /VILI 73/ Villedra, J. e Wieringa, R.F. (eds.): "Computer-Aided Design", North-Holland, 1973.
- /VERNA 84/ Vernadat, F.B.: "A Commented and Indexed Bibliography on Data Structuring and Data Management in CAD/CAM: 1970 to MID-1983".
- /WG10 77/ Wiederhold, G.: "Database Design", McGraw-Hill, 1977