



**XII CONGRESSO NACIONAL DE
PROCESSAMENTO DE DADOS-SUCESU**

PALÁCIO DAS CONVENÇÕES-PARQUE ANHEMBI-S. PAULO-DE 8 A 12 DE OUTUBRO DE 1979

**XII CONGRESSO
NACIONAL DE
PROCESSAMENTO
DE DADOS**

SUCESU

004.06
C749
1979

PAULO OUTUBRO 1979

XII CONGRESSO NACIONAL DE PROCESSAMENTO DE DADOS

ANAIS DO XII CNPD

Centro de Convenções do
Parque Anhembi-São Paulo
Outubro de 1979

COMISSÃO ORGANIZADORA

Salvador Perrotti

Boris Liberman

Jorge Coimbra

Múcio Álvaro Doria

Rubens Lessa Vergueiro Filho

Fukuhara Takatika

Gabriel Liebesny

Eduardo Caldas Pereira

Anita Woiler

Renato Antônio Mazzola

Raul Isiris

O MAPEAMENTO DA ESPECIFICAÇÃO CONCEITUAL EM BANCO DE DADOS GERENCIADOS PELO ADABAS

Rubens N. Melo
Leonardo Lellis Leite
PUC/RJ

Palavras-chave: Sistemas de informação, Especificação conceitual, Banco de Dados, Esquema conceitual, Esquema interno, Mapeamento conceitual/interno.

RESUMO:

Com o objetivo de se conseguir uma metodologia para o mapeamento da especificação conceitual de um Banco de Dados (BD) para um esquema de BD no nível dos SGBDs comerciais, este é mais um trabalho de uma série que trata desse problema e que nesse caso considera o SGBD ADABAS.

Inicialmente são apresentadas as idéias básicas da especificação conceitual de sistemas de informação e uma linguagem para especificação conceitual de Banco de Dados. Em seguida algumas características relevantes do sistema ADABAS são abordadas e, finalmente, é sugerida uma maneira de mapear a especificação conceitual do Banco de Dados do sistema para um esquema interno usando o ADABAS.

1. INTRODUÇÃO

Segundo um estudo recente da CAPRE sobre a utilização de Banco de Dados no Brasil, a maioria das empresas que usam Sistemas de Gerência de Banco de Dados (SGBDs) não adotam ainda uma política de integração dos seus sistemas a nível da empresa.

Na maioria das vezes apenas os sistemas antigos foram *convertidos para Banco de Dados* onde a mudança principal foi a substituição dos métodos de acesso convencionais pelo *software* de Banco de Dados.

Um dos grandes empecilhos para a reformulação ou redefinição dos sistemas da empresa procurando integrá-lo em torno de um Banco de Dados comum, é a falta de uma linguagem adequada que permita a participação efetiva dos usuários e dos administradores na especificação do sistema integrado.

Além disso nos SGBDs disponíveis comercialmente, nota-se uma certa deficiência nos esquemas de definição de dados adotados, no que diz respeito a sua compreensão (e efetiva participação na especificação) por parte dos usuários finais.

A especificação conceitual de sistemas de informação (1,2) visa a participação ativa e construtiva do usuário

final na especificação do sistema de informação a ser implementado.

Este trabalho aborda o problema da obtenção do esquema interno de Banco de Dados (3), a ser gerenciado pelo ADABAS (4), a partir de sua especificação conceitual, feita em uma linguagem específica desenvolvida em (6).

Um resumo das idéias básicas de Especificação Conceitual de Sistemas de Informação apoiados em Banco de Dados e dos modelos (conjunto de conceitos) para especificação do esquema conceitual de BDs é descrito na Seção 2. Complementando esta Seção é apresentada uma (proposta de) linguagem para especificação do esquema conceitual de BDs.

Supondo que esta especificação conceitual tenha sido analisada (quanto a correção e completesa) e armazenada em um Dicionário/Diretório de dados, este trabalho sugere na Seção 4 um método para mapear a especificação conceitual dos dados para um esquema interno de um BD a ser gerenciado pelo ADABAS.

Visando a auto-suficiência do trabalho, na Seção 3 é apresentado um resumo sobre as estruturas de arquivos do sistema ADABAS assim como dos métodos oferecidos para definição do esquema interno do Banco de Dados.

2. INTRODUÇÃO A ESPECIFICAÇÃO CONCEITUAL DE SISTEMA DE INFORMAÇÃO

Este trabalho, é segundo de uma série, na qual estão sendo investigados os caminhos a serem seguidos, no sentido da obtenção de uma linguagem para especificação conceitual de sistemas de informação apoiados em Banco de Dados, bem como uma metodologia para o mapeamento desta especificação para uma especificação a nível interno do Banco de Dados.

No primeiro trabalho (7) foram introduzidos os fundamentos da especificação conceitual, um esquema para o modelo conceitual de Banco de Dados (ENTIDADES, ATRIBUTOS & RELAÇÕES) e uma proposta de mapeamento de uma especificação conceitual de Banco de Dados (feita utilizando o esquema apresentado) em esquemas internos de Banco de Dados gerenciados pelo TOTAL.

Dando continuidade a esta linha de estudos, o presente trabalho:

- enfatiza a importância de especificações conceituais de Banco de Dados;
- apresenta proposta de linguagem para especificação do esquema conceitual do Banco de Dados;
- mostra como pode ser feito o mapeamento, de esquema conceitual (especificado com a linguagem apresentada), em um esquema interno de um Banco de Dados a ser gerenciado pelo ADABAS.

2.1 *Conceitos Básicos*

Analisando os aspectos teóricos que apresentam os objetivos e os benefícios a serem alcançados com a utilização de Sistemas de Banco de Dados, nota-se nitidamente, que alguns dos principais benefícios que podem ser alcançados, são decorrências diretas da obtenção da integração e do controle centralizado dos dados da empresa.

O valor deste controle centralizado é proporcional ao *nível de abrangência e integração* das informações contidas no Banco de Dados.

Analisando os caminhos que estão sendo seguidos pelas empresas, na implantação de sistemas de informação apoiados em Banco de Dados, nota-se, em muitos casos, que é o analista responsável por determinada aplicação quem especifica o modelo do Banco de Dados a ser utilizado.

Este enfoque é obviamente indesejável, uma vez que dificulta (e em certos casos nem considera) a integração das informações a nível de empresa, que é o nível de integração mais valioso a ser alcançado.

Na realidade, um dos principais motivos deste estado de coisas, é a não participação dos usuários finais, dos gerentes da empresa que tenham a visão geral e integrada da empresa no processo de especificação conceitual do Banco de Dados da empresa. Esta não participação se deve em grande parte, a falta de ferramenta adequadas para tal, uma vez que os esquemas de definição dos Banco de Dados, utilizados pelos Sistemas de Gerência de Banco de Dados (SGBDs) disponíveis no mercado, são mais orientados para os técnicos de processamento de dados do que para os usuários finais dos dados, propriamente ditos.

Levando-se em conta que os sistemas de informações que utilizam SGBD, tem se tornado cada vez mais abrangentes e, com isto, os dados que eles manipulam passam a ser um valioso recurso, que deve ser compartilhado por diversos setores da empresa, a participação do usuário final no processo de especificação do modelo a ser adotado pelo Banco de Dados da empresa, torna-se um fator imprescindível.

Mesmo as novas ferramentas tais como os sistemas de Dicionários de Dados já disponíveis no mercado e que muitas vezes são utilizados em ambientes de Banco de Dados visando obter uma melhor integração dos sistemas de empresa, carecem de uma metodologia adequada de uso.

A necessidade de ferramenta que permitam a maior participação dos administradores no processo de especificação, pode ser considerada como uma das principais motivações dos estudos sobre a especificação conceitual de sistemas de informações apoiados em Banco de Dados.

A formalização da especificação conceitual, em termos de uma linguagem especificamente criada para esta finalidade, pode tornar possível a transformação sistemática dos requisitos dos usuários em problemas para técnicos em processamento de dados, possibilitando que o Banco de Dados seja visto como uma fonte de informações, logicamente interrelacionadas, sem ser necessário o conhecimento de sua estrutura física.

Os modelos lógicos de especificação devem ser flexíveis, no sentido de facilitarem a obtenção de esquema conceituais que permitam ao usuário, utilizar o Banco de Dados num modelo orientado para a realidade, em termos que sejam compatíveis com a sua própria concepção do mundo real. É importante que este modelo possa evoluir naturalmente (ser modificado) no sentido de acompanhar a dinâmica da realidade que modela.

De um modo geral a integração dos dados de toda empresa leva a criação de modelos conceituais que envolvem diversas entidades do mundo real, que embora façam parte da empresa, não necessitam ser consideradas em sua totalidade em cada uma das aplicações

(subsistemas) que devem compartilhar o Banco de Dados.

Este fato nos leva a duas considerações importantes sobre o modelo a ser adotado na especificação conceitual do Banco de Dados. A primeira é que a formulação do modelo deve ser simples para que não se complique a representação de certas situações do mundo real, que são intrinsecamente complexas. A segunda é que o modelo deve servir de base para todos os tipos de visões externas (subconjunto das entidades e suas relações representadas no modelo) que são necessárias para as diversas aplicações. Cada visão externa deve poder ser construída levando em conta apenas as entidades e relações que são relevantes (necessárias) a ampliação (ou conjunto de aplicações) que a utilizarem.

É necessário ainda, por questões de consistência lógica, que a modelagem seja feita utilizando-se uma linguagem que não permita ambiguidades e ao mesmo tempo seja capaz de representar todas as situações do mundo real que devem ser tratadas.

Sintetizando podemos dizer que a especificação conceitual constitui-se numa mudança de enfoque que se tornou necessária para a evolução da utilização do Banco de Dados a nível de empresa, e não mais a nível de aplicações isoladas. Tradicionalmente, os modelos utilizados eram mais orientados para os aspectos internos dos sistemas, o novo enfoque caracteriza-se por ser mais dirigido ao usuário final.

Assim sendo o esquema conceitual:

- i) permite que os diversos usuários finais do Banco de Dados tenham uma compreensão (uma "visão") comum do conteúdo do Banco de Dados.
- ii) fornece ao implementador uma descrição de que dados devem ser armazenados no Banco de Dados.

Para que estes objetivos sejam alcançados, torna-se necessário que o modelo abstrato utilizado para o esquema conceitual satisfaça as seguintes condições:

- seja fácil de formular e compreender;
- seja completo;
- sirva de base para todos os tipos de visões externas necessárias à empresa;
- possa evoluir (se modificar) facilmente;
- seja ortogonal, ou seja, diferentes conceitos do modelo abstrato devem ser expressos por diferentes conceitos no esquema conceitual, e a cada conceito do modelo abstrato deve corresponder somente um conceito no esquema conceitual.

2.2 Uma Linguagem para Especificação Conceitual

Diante de necessidade da utilização de esquemas conceituais para Banco de Dados, têm sido feitos estudos tentando estabelecer o conjunto de conceitos mais adequados para a especificação do esquema conceitual.

Diversas propostas para este conjunto de conceitos são encontradas na literatura, dentre elas podemos citar Biller e Neulhold (8), Nijssen (9), Benci et al (10) e Grotenhuis (11).

A partir da análise destes trabalhos, pode-se concluir que existem basicamente dois tipos de modelos:

- i) os modelos baseados em dois conceitos que são: entidades e relações;
- ii) os modelos baseados em três conceitos que são: entidades, relações e atributos.

No entanto, estes dois tipos de modelos não possuem diferenças muito significativas (a maioria das diferenças residem na terminologia adotada) e acreditamos que dentro de pouco tempo será estabelecido um consenso sobre o assunto, para possibilitar uma correspondência entre os conceitos utilizados que permita a transformação entre modelos diferentes. Esta correspondência é fundamental para que se realize a coexistência entre os modelos e, por conseguinte, entre os sistemas de Banco de Dados.

Analisando os aspectos do mundo real que normalmente são descritos e utilizados em sistemas de informação, concluímos que *uma certa classe de aplicações* pode ser descrita através de ENTIDADES que possuem propriedades elementares conhecidas como ATRIBUTOS e de RELAÇÕES entre estas entidades.

Uma descrição deste modelo pode ser encontrada em (7). Na linguagem que apresentaremos este modelo foi expandido para aceitar ATRIBUTOS simples, compostos e com possibilidade de múltiplas ocorrências.

A seguir apresentaremos esta linguagem. Inicialmente será apresentado o alfabeto e as convenções utilizadas, depois a forma genérica e a descrição do esquema para especificação conceitual do Banco de Dados.

- Alfabeto

Caracteres alfabéticos — A a Z

Caracteres numéricos — 0 a 9

Caracteres especiais — — . , ; () :

- **Convenções:**

As palavras são formadas com os caracteres alfabéticos, numéricos e o hífen, devem iniciar com um caracter alfabético e podem ser dos seguintes tipos:

- palavras chaves do sistema, que são apresentadas em letras maiúsculas e devem ser utilizadas exatamente como são apresentadas;
- parâmetros do usuário, que são apresentados em letras minúsculas e são codificados de acordo com a aplicação que estão sendo definida e devem ter no máximo 30 caracteres;
- Os números devem ser inteiros sem sinal;
- O início de um comando é indicado por uma palavra chave e o fim por um ponto. Quando em um comando existem diversos parâmetros do usuário eles devem vir separados por vírgulas. O ponto e vírgula é utilizado para separar dois grupos de parâmetros relacionados. A utilização de espaço é livre.
- Símbolos:
- parênteses — devem ser utilizados na forma em que aparecem no texto.
- chaves — não devem ser utilizadas. No texto indicam que uma das opções incluídas entre as chaves deve ser especificada.
- colchetes — não devem ser utilizados. No texto indicam que o ítem é opcional. Quando existe mais que uma opção e uma delas estiver sublinhada indica que ela será assumida no caso da não especificação do ítem.
- reticências — não devem ser utilizadas. No texto indicam que o ítem pode ser utilizado quantas vezes se fizer necessário.
- Linguagem para definição do esquema conceitual

Esquema-Conceitual-do-BD nome-do-banco-de-dados.

[*Autor* nome-do-autor.]

[*Chave-de-Acesso* chave-de-acesso.]

[*Observações* comentários.]

Definição-das-Entidades.

Entidade nome-da-entidade [*Sinônimos* sinônimo-1;...]

Atributos [nível] nome-do-atributo-1 $\left[\begin{matrix} 1 \\ (\text{número-de-ocorrências}) \\ N \end{matrix} \right]$

[*Sinônimos* sinônimo-1; ...:sinônimo-n] [; ...]

Identificador nome-do-atributo-1 [, ...]

[*Ocorrências* número-estimado-de-ocorrências.]

[*Crescimento* estimativa-de-crescimento.]

[*Observações* comentários.]

Definição-das-Relações

Relação nome-da-relação [*Sinônimos* sinônimo-1, ...]

Entidades [*Relacionadas*] nome-da-entidade-1, ..., nome-da-entidade-n.

Conectividade $\left\{ \begin{matrix} 1 \\ (\text{número-de-ocorrências}) \\ N \end{matrix} \right\} : \dots : \left(\begin{matrix} \text{número-de-ocorrência} \end{matrix} \right)$

[*Atributos Associados* [nível] nome-do-atributo-1

$\left[\begin{matrix} 1 \\ (\text{número-de-ocorrências}) \\ N \end{matrix} \right]$

[*Sinônimos* sinônimo-1, ... , sinônimo-n] [; ...] .]

[*Observações* comentários.]

FIM-DO-ESQUEMA-CONCEITUAL.

Nesta especificação temos três seções: a geral, a de definição de entidades e a de definição de relações.

A seção geral inicia com o comando de definição do nome do Banco de Dados que deve ser o primeiro comando da definição de um esquema conceitual. Opcionalmente pode ser especificada uma entrada contendo o nome do autor, uma entrada contendo uma chave de acesso a definição do esquema conceitual e uma entrada de comentários.

A seção de definição de entidades inicia pelo comando DEFINIÇÃO-DAS-ENTIDADES e deve seguir a seção geral. A especificação de cada entidade deve iniciar pelo comando ENTIDADE que define o nome da entidade e opcionalmente sinônimos para este nome. Os demais comandos referentes a esta entidade podem ser especificados em qualquer sequência. No comando ATRIBUTO devem ser especificados os nomes dos atributos que a entidade possui é permitida a estruturação dos atributos através da especificação de níveis; quando um determinado atributo pode ocorrer mais que uma vez para cada ocorrência da entidade, deve ser indicado o número de ocorrências possíveis, no caso deste número ser indeterminado definidos nomes sinônimos para os atributos.

O comando IDENTIFICADOR deve conter uma lista de um ou mais nomes de atributos que juntos identificam univocamente uma ocorrência da entidade.

Complementando a definição de uma entidade, podem ser fornecidos o número estimado de ocorrências e a taxa estimada de crescimento, estes parâmetros são utilizados na emissão de um relatório de apoio a especificação do esquema interno, sendo que a forma de especificação da taxa de crescimento é escolhida pelo usuário (por exemplo: número de entidades adicionais por mês ou aumento percentual por mês).

A seção de definição de relações inicia pelo comando DEFINIÇÃO-DAS-RELAÇÕES e deve seguir a seção de definição das entidades. A especificação de cada relação deve iniciar pelo comando RELAÇÃO que define o nome da relação e opcionalmente seus sinônimos. Os demais comandos referentes a esta relação podem ser especificados em qualquer sequência. No comando ENTIDADES devem ser especificados os nomes das entidades que participam da relação. No comando CONECTIVIDADE deve ser estimado a proporção de ocorrências de cada entidade participante da relação na mesma ordem em que foram definidos os nomes das respectivas entidades no comando ENTIDADES; para indicar um número indeterminado maior que 1 deve ser usado o símbolo N. Quando existirem dados associados a relação eles devem ser especificados no comando ATRIBUTOS de forma análoga a utilizada para definição de atributos de entidades, descrita anteriormente.

O fim da definição de um esquema conceitual é indicado pelo comando FIM-DO-ESQUEMA-CONCEITUAL.

A especificação de um esquema conceitual deve ser analisada por um programa que verifica sua correção, emite um relatório contendo a imagem da definição e possíveis mensagens de erro e quando a definição está correta a armazena no dicionário/diretório de dados do sistema.

3. DBMS ADABAS

O Adaptable Data Base System ADABAS (4,5) é um SGBDs que utiliza arquivos invertidos para acesso e relacionamento entre arquivos. Não existe uma estrutura lógica-pré-estabelecida; cabe ao usuário a tarefa de estruturação, devendo para tanto especificar todos os relacionamentos lógicos desejados.

O acesso aos dados é feito através de programas de aplicação utilizando chamadas a uma rotina padrão, que faz a comunicação entre o programa e o ADABAS; ou através de uma linguagem de recuperação (ADASCRIP) que permite ao usuário final acesso direto aos dados armazenados.

O sistema possui uma linguagem de definição de dados

própria, cujos comandos são divididos em dois grupos, comandos definição de arquivos e comandos de definição de relacionamento (acoplamentos na terminologia do ADABAS) e oferece dois programas para geração das tabelas de descrição do Banco de Dados a partir das especificações feitas através dos comandos da linguagem de definição. Estas tabelas são utilizadas pelo ADABAS para atender os comandos colocados nos programas de aplicação. Para o atendimento dos comandos feitos através da linguagem de recuperação própria do sistema, deve ser criada uma tabela para mapeamento de nomes externos em nomes internos.

A seguir analisaremos as características funcionais do ADABAS no que diz respeito ao esquema de definição e a organização de arquivos.

3.1 Esquema de Definição

A descrição do Banco de Dados é feita através de um conjunto de parâmetros tratados por dois utilitários do sistema (*Loader e Couple*). As visões locais são definidas dentro dos programas de aplicação através da especificação da formatação dos registros lógicos a serem utilizados (*Format Buffer*).

O modelo adotado pelo ADABAS é o seguinte: o Banco de Dados é formado por um conjunto de arquivos lógicos (*Files*) relacionados (*Coupled*) através de ligações binárias de conectividade muitos-para-muitos. A definição de um *File* é feita através de um conjunto de registros, cada um dos quais contendo a descrição de um dos campos (*Fields*) do *File*. A definição de um campo é feita através da especificação dos seguintes parâmetros posicionais:

nível, nome, tamanho, formato, opções...

onde:

nível — especifica o nível do campo dentro de uma estrutura hierárquica (1 nível raiz, número máximo de níveis: 7)

nome — nome do campo (*2bytes*)

tamanho —

tamanho padrão do campo, tamanho que será utilizado quando um programa de aplicação não indicar o tamanho do campo. (Especifica também o tamanho do campo para carga inicial via utilitário).

Formato —

formato dos campos de nível elementar, a ser utilizado como formato de armazenamento (e carga inicial via utilitário).

Opções —

especificação de condições especiais do campo, podendo ser especificado:

DE — descritor (possuirá lista invertida)

FI — tamanho de armazenamento fixo (sem compressão)

NU — supressão do campo quando o valor for:

. zero para campos numéricos

. brancos para campos alfanuméricos.

MU — campo com múltiplos valores (máximo de 191)

PE — grupo periódico (máximo de 99).

o utilitário *Loader* processa esta definição e gera uma tabela de descrição para o *File*. Esta tabela será colocada no *Associator* junto as tabelas de descrição dos demais *Files*. Este utilitário executa também a função de definição das características físicas, formatação e carga inicial do *File*. Fisicamente um Banco de Dados ADABAS é formado por três arquivos: *ASSOCIATOR* (que contém as tabelas de descrição dos arquivos, as listas invertidas e o conversor ISNs em endereços), *DATA* (que contém os dados) e *WORK* (um arquivo de trabalho). O utilitário *Loader* é dividido em três rotinas:

- análise da definição lógica do *File*, edição e compressão dos dados para carga inicial (ADAWAN);
- alocação e formatação de arquivos físicos (só realizada na carga do primeiro *File*), definição das características físicas do *File* (número de registros, espaço em cilindros e percentagem de área para crescimento), carga da tabela de definição e carga dos dados (ADALDI);
- carga das listas invertidas dos campos descritores (ADALD2)

Para estabelecer os relacionamentos entre *Files* existe o utilitário *Couple* que recebe como entrada um registro contendo a seguinte especificação:

FILE número-do-file (nome-do-campo) WITH FILE número-do-file (nome-do-campo)

Este utilitário estabelece o relacionamento lógico (colocando a indicação nas tabelas de descrição dos arquivos) e físico (criando e carregando as listas invertidas que mantêm o relacionamento).

A execução do processo de definição do Banco de Dados é feita através de dois procedimentos, um de definição e carga do *File* e outro de definição e implantação dos relacionamentos *Couple*.

O processo de definição e carga de um *File* é composto de três passos:

1) ADAWAN

- analisa e carrega a definição
- processa, editando e comprimindo os dados a serem carregados

2) ADALD1

- inicializa o arquivo em disco a partir de parâmetros lidos de cartões
- carrega os dados

3) ADALD2

- cria listas invertidas

É interessante observar que este processo não separa a tarefa de definição da tarefa de carga, ou seja o processo de definição lógica das estruturas internas é feito junto com o processo de carregamento físico do arquivo.

O processo de definição e implementação dos relacionamentos *Couple* é composto de dois passos:

1) ADACP1

- analisa os parâmetros de *Couple*
- lê o arquivo *DATA* criando um arquivo de trabalho

2) ADACP2

- ordena o arquivo de trabalho
- cria os dados sobre o *Couple*
- carrega os dados sobre *Couple* no *ASSOCIATOR*

Este processo de *Coupling* está sujeito a algumas restrições (apresentadas no manual UTILITIES (5), entre estas uma restringe o processo de modelagem, e consiste no fato de dois *Files* só podem ser acoplados (*Coupled*) através de um par de descritores.

3.2 Organização de arquivos

A estrutura lógica de dados utilizada pelo ADABAS é a estrutura de redes. Esta estrutura é composta por um

conjunto contendo de um a duzentos e cinquenta e cinco *Files*, sendo que cada *File* pode estar relacionado (*Coupled*) de zero a oitenta outros *Files*. Os relacionamentos mantidos pelo ADABAS são binários de conectividade muitos-para-muitos.

Ao nível físico esta estrutura lógica é implementada através de dois tipos de arquivos físicos:

- o arquivo que contém os dados propriamente ditos, DATA, que é um arquivo de acesso direto (BDAM), com blocos de tamanho fixo, sendo que o tamanho dos blocos é pré-estabelecido pelo ADABAS para cada tipo de dispositivo, sendo de um modo geral da ordem de 3k bytes. Cada bloco contém um número de registros, este número varia de bloco para bloco de acordo com o *File* ao qual pertencem os registros e com o tamanho de cada registro, que por sua vez depende do fator de compressão que foi aplicado de acordo com os valores dos campos que o compõem.
- o ASSOCIATOR arquivo que contém as listas invertidas, o conversor de números internos (ISNs) em endereços e as tabelas de definição de *Files*. Este arquivo é de acesso direto (BSAM), com blocos de tamanho fixo, sendo o tamanho pré-estabelecido de acordo com o dispositivo, sendo da ordem de 1,5k bytes. Cada lista invertida é formada pelo valor do descritor, número de registros que contém aquele valor e ISNs de cada um dos registros. O conversor de endereços é um vetor unidimensional, contendo endereços físicos de blocos (3k bytes).

4. O MAPEAMENTO DA ESPECIFICAÇÃO CONCEITUAL PARA UM ESQUEMA INTERNO DO BANCO DE DADOS A SER GERENCIADO PELO ADABAS.

4.1 O caminho a ser seguido

Nesta Seção abordamos o problema do mapeamento de uma especificação conceitual de um Banco de Dados, feita utilizando a linguagem de especificação conceitual (apresentada na Seção 2.2) para um esquema interno de Banco de Dados gerenciado pelo ADABAS (4,5).

Quando em (7) analisamos o problema do mapeamento da especificação conceitual para esquema interno de Banco de Dados gerenciado pelo TOTAL (12) seguimos um caminho proposto em (2) que consistia em utilizar a linguagem PSL (*Problem Statement Language*) do sistema PSL/PSA (13, 14) como linguagem para especificação conceitual. A especificação feita em PSL era analisada e armazenada em um Banco de Dados próprio e depois o processo responsável pelo mapeamento gerava o esquema interno para o SGBD a ser utilizado

(no caso considerado em (7) o SGBD era o TOTAL).

Embora a PSL tenha facilidades muito boas para outros aspectos da especificação conceitual do sistema como um todo por exemplo para a especificação dos aspectos dinâmicos do sistema, ela não é adequada em certos aspectos que dizem respeito a especificação do Banco de Dados.

Diante destas considerações no presente trabalho seguiremos o mesmo enfoque porém utilizando a linguagem da Seção 2.2 para fazer a especificação conceitual do Banco de Dados. Esta especificação é analisada por um programa do SGBD que verifica sua correção e completeza e a armazena no dicionário/diretório de dados do sistema. A partir desta definição armazenada, e da especificação das regras de mapeamento para o SGBD escolhido (fornecidas pelo administrador do Banco de Dados), o programa responsável pela geração do esquema interno faz o mapeamento do esquema conceitual, para o esquema interno do SGBD escolhido para a implementação dos sistemas de aplicação.

É interessante observar que este procedimento é semelhante ao proposto pela ANSI/SPARC em seu relatório sobre a arquitetura do SGBD (3). Esta arquitetura, no que diz respeito a definição dos esquemas de dados, consiste na existência de três tipos de esquemas de definição, o esquema conceitual que é especificado pelo administrador de dados da empresa, os esquemas externos que são especificados pelos responsáveis pelas aplicações e o esquema interno que é especificado pelo administrador do Banco de Dados.

A figura 1 ilustra o relacionamento entre os esquemas e administradores na arquitetura proposta pelo ANSI/SPARC.

A diferença entre o caminho proposto no presente trabalho e arquitetura do ANSI/SPARC reside no fato, do processo de geração do esquema interno, aqui proposto, ser automático, cabendo ao administrador do Banco de Dados definir apenas as regras do mapeamento (conforme veremos na Seção seguinte para o ADABAS) enquanto que no procedimento proposto pelo ANSI/SPARC o ABD manualmente define o esquema interno guiado pelo que foi especificado no esquema conceitual.

É importante notar que deste modo os controles definidos a nível conceitual só poderão ser impostos por procedimentos e disciplinas estabelecidas pelo ABD e não serão automaticamente controlados pelo sistema.

4.2 As Regras para o Mapeamento

Nesta Seção estamos interessados em propor as regras a serem seguidas para o mapeamento da especificação conceitual do Banco de Dados ou seja das ENTIDADES,

seus RELACIONAMENTOS e seus ATRIBUTOS para o esquema interno de um Banco de Dados ADABAS, que conforme vimos na Seção 3, é composto de *Files* cujos registros podem ser estruturados em níveis formados por *Fields* sendo a associação entre *Files* feita através de *Couplings* entre eles.

De maneira geral os *Files* são candidatos naturais para a implementação de entidades. As relações de grau maior que dois com atributos associados também são parcialmente implementadas por *Files*. Os *Couplings* são adequados para o mapeamento das relações binárias com conectividade muitos-para-muitos (N : N) e com atributos associados. Porém vejamos a seguir com um pouco mais de detalhes as alternativas desses mapeamentos.

4.2.1 Mapeamento das Entidades e seus Atributos

- Os Atributos

No modelo conceitual as ENTIDADES tem ATRIBUTOS simples ou compostos que ocorrem uma única vez ou múltiplas vezes. Os *Fields* do ADABAS podem representar todos esses casos pois a definição de um registro pode incluir grupos (ATRIBUTOS compostos) "periódicos" que podem ter até no máximo 99 ocorrências num registro lógico (ATRIBUTOS compostos que ocorrem múltiplas vezes). Além disso os campos múltiplos permitidos em ADABAS representam adequadamente os ATRIBUTOS simples de ocorrência múltipla.

O ADABAS identifica os *Files* através de um número sequencial que é atribuído a cada um deles. Assim sendo a tabela que contém informações sobre o mapeamento, deverá conter a associação do nome de cada tipo de entidade, assim como de seus sinônimos, com o respectivo número de arquivo para o qual foi feito o mapeamento. Além disso essa tabela deve conter outras informações que permitam fazer a correspondência entre os outros nomes usados no nível conceitual e as abreviações (exageradas) usadas no nível do esquema interno. Por exemplo para cada nome de atributo no nível conceitual deve ter a abreviação de 2 caracteres usada no esquema ADABAS.

- A Cardinalidade das Entidades

O parâmetro OCORRÊNCIAS número-estimado-de-ocorrências será mapeado para o parâmetro de cartão tipo D do ADALD1 (5) que contém o número de registros em milhares.

O parâmetro CRESCIMENTO estimativa-de-crescimento, será mapeado para o parâmetro D = X do cartão D, onde X representa a percentagem de espaço a ser reservada para o crescimento dentro dos blocos do arquivo DATA, assim como para o parâmetro A = y do mesmo cartão D, que representa a percentagem de espaço a ser reservada

para crescimento dentro dos blocos do ASSOCIATOR.

- O Identificador

O conceito de IDENTIFICADOR não é adequadamente representado no ADABAS. Entretanto como uma entidade é referenciada pelo seu identificador, pelo menos a opção DESCRITOR deve ser atribuída ao campo (ou grupo) que representa o identificador da entidade.

4.2.2 Mapeamento das Relações e seus Atributos Associados

Existem várias alternativas para o mapeamento de relações, dependendo da conectividade da relação.

i) Relações Binárias de Conectividade (1 : N)

- Sem atributos associados

a) São naturalmente implementadas por acoplamento (*Coupling*) entre os *Files* que representam as ENTIDADES. O campo representante do IDENTIFICADOR da ENTIDADE "pai" da relação deve ser acrescentado como campo do *File* que representa a ENTIDADE subordinada. O acoplamento é definido por este campo comum.

b) Outra alternativa é implementar ambas as ENTIDADES num só *File* representando a ENTIDADE subordinada como um grupo periódico da ENTIDADE pai. Nesse caso a conectividade pode ir até 1 : 99. Note-se que a ENTIDADE subordinada pode ser referenciada isoladamente pois o ADABAS permite a recuperação por um campo de um grupo periódico.

- Com atributos associados

Nesse caso os atributos podem ser "transferidos" a ENTIDADES pai e representados como *Fields* do *File* pai.

ii) Relações Binárias de Conectividade (N : N)

- Com atributos associados

a) Os atributos associados são representados como campo (ou grupo) comum em ambos os *Files* e que serve de base para um *Coupling* entre os *Files*.

b) Através de 3 *Files* e 2 *Couplings* (vide relações de grau maior que 2)

- Sem atributos associados

Como a conectividade é N : N parece que a única saída é via *Coupling* e como este precisa de algum campo comum o que se pode fazer é criar artificialmente

um campo comum como se existisse um ATRIBUTO associado a relação. Com este artifício a solução é similar a do caso anterior.

iii) Relações de grau N maior que 2

- Com atributos associados

- Cada ENTIDADE relacionada é mapeada para um *File*.

- Um *File* intermediário com campos correspondentes aos atributos associados juntamente com os IDENTIFICADORES de cada uma das ENTIDADES relacionadas.

- N acoplamentos. Cada *File* correspondente a cada ENTIDADE deve ser acoplado com o *File* intermediário via o campo comum entre eles que é o que representa o IDENTIFICADOR da ENTIDADE em questão.

- Sem atributos associados

A solução é a mesma exceto que o *File* intermediário de ligação só terá campos correspondentes aos IDENTIFICADORES das ENTIDADES relacionadas.

iv) Outros casos

- Hierarquias de ENTIDADES

Esse é o caso geral do qual o i) é um caso particular. Dentro dos limites do ADABAS ou seja: no máximo 7 níveis de hierarquia, máxima de 99 ocorrências para grupos periódicos e máxima de 191 valores para campos múltiplos, podemos implementar uma relação hierárquica de ENTIDADES como um só *File*.

- Relações com conectividade (1 : 1 : : 1)

Esses são casos mais especiais ainda e podem ser facilmente ser implementadas como um só *File*.

4.2.3 Observações

Nota-se que as facilidades do ADABAAS favorecem mais o mapeamento das ENTIDADES. A facilidade de acoplamento (*Coupling*) como está implementada no ADABAS, impõe duas restrições ao mapeamento das relações.

- 1) a primeira, que consideramos seja mais forte, reside no fato de o ADABAS não permitir mais que um acoplamento entre dois *Files* o que impede o mapeamento de mais que uma relação entre duas determinadas entidades. A razão de considerarmos esta restrição como sendo uma "forte restrição" baseia-se no fato de não serem raras as "situações

do mundo real" em que encontramos duas entidades relacionadas através de mais que um tipo de relação.

- 2) a segunda restrição reside no fato do ADABAS exigir que para que este tipo de ligação seja mantido, exista um *Field* comum aos dois *Files*; o que obviamente não é o caso mais comum, quando duas entidades do mundo real estão relacionadas.

5. CONCLUSÕES E FUTUROS TRABALHOS

O objetivo deste trabalho é dar mais um passo para se chegar a uma metodologia de mapeamento da especificação de Banco de Dados a nível conceitual para um esquema interno no nível dos SGBDs comumente disponíveis no mercado e usados na prática. Os primeiros trabalhos (2,7) se referiram a SGBDs do mesmo nível de conceitos que o ADABAS. Nessa pesquisa o SGBD utilizado é irrelevante, o que se procura são subsídios para uma sistemática no mapeamento.

Entre as limitações notadas nessa análise podemos mencionar a dificuldade de representação do conceito de IDENTIFICADOR e de múltiplas RELAÇÕES entre as mesmas duas ENTIDADES.

Vários pontos ainda precisam ser estudados para atingir o principal objetivo. Entre eles pode-se mencionar:

- considerar no mapeamento conceitual/interno as informações sobre tipos de acesso desejados as entidades conforme especificações nos esquemas externos;
- obter uma forma genérica padronizada para representação do esquema interno, visando a automatização do mapeamento conceitual/interno.
- considerar os aspectos dinâmicos da especificação conceitual do sistema de informação como um todo e integrá-los com mapeamento conceitual/interno.

CURRICULUM VITAE

RUBENS NASCIMENTO MELO

Dr. Sc ITA 1976

Prof. Associado do Departamento de Informática da PUC/RJ

Áreas de Interesse:

Sistemas de Informação; Banco de Dados e Engenharia de Software.

LEONARDO LELLIS LEITE

M. Sc PUC 1978

Prof. do Departamento de Informática da PUC/RJ
(Projeto 15)

Analista de Sistemas da NUCLEBRÁS

Áreas de Interesse:

Sistemas de Informação e Banco de Dados.

BIBLIOGRAFIA

- 1 — SUNDGREN, B. *Theory of Data Bases*. New York, Petrocelli, 1975.
- 2 — GUSMÃO, E.D. *Especificação Conceitual de Sistemas de Informação Apoiados em Banco de Dados*. Rio de Janeiro, PUC, Departamento de Informática, Tese de Mestrado, 1976.
- 3 — ANSI/X3/SPARC. Study Group on Data Base Management Systems *Interim Report Feb 75*.
- 4 — SOFTWARE AG. *ADABAS Reference Manual*. Darmstadt, Nov 1973.
- 5 — SOFTWARE AG. *ADABAS Utilities Manual*. Darmstadt, 1974.
- 6 — LELLIS, L. *Análise e Projeto da Arquitetura para um Sistema de Gerência de Banco de Dados*. Rio de Janeiro, PUC, Departamento de Informática. Tese de Mestrado, Dez 1978.
- 7 — MELO, R. & LELLIS, L. O Mapeamento da Especificação Conceitual em Banco de Dados Gerenciados pelo TOTAL. PUC, Departamento de Informática, Monografia em Ciência da Computação nº 14/77.
- 8 — BILLER, H. & NEUHOLD, E.J. *Semantics of Data Bases*. Stuttgart, Univ. Stuttgar, Institut für informatk, TR 03/76.
- 9 — NIJSSEN, G.M. A Gross Architecture for the Next Generation DBMS In: NIJSSEN, G.M. *Modelling in DBMS; Proceedings of trhe IFIP TC-2 Working Conference in Frendestadt, Jam 76*. Amsterdam, North-Holland (1976)
- 10 — BENCI et al. Concepts for design of a conceptual schema In: (9)
- 11 — GROTENHUIS, G. & BROEK, J. Concepts for Modelling Information. In: (9)
- 12 — Os TOTAL Referece Manual. Ohio. Cincon, 1976.
- 13 — TEICHROEW, D. & HERRSHEY III. An Introducion to PSL/PSA. Michigan ISDOS Working Paper, May 1975.
- 14 — HERSHEY et al. Proble Statement Language Version 3. *O Languge Reference Manual*. Michigan, Univ. of Michigan ISDOS Working Paper, May 1975.
- 15 — YORMARK, B. The ANSI/X3/SPARC/SGBDMS Architecture: In: JARDINE, D.A. *The ANSI/SPARC DBMS Model*. Amesterdam, North-Holland, 1977.

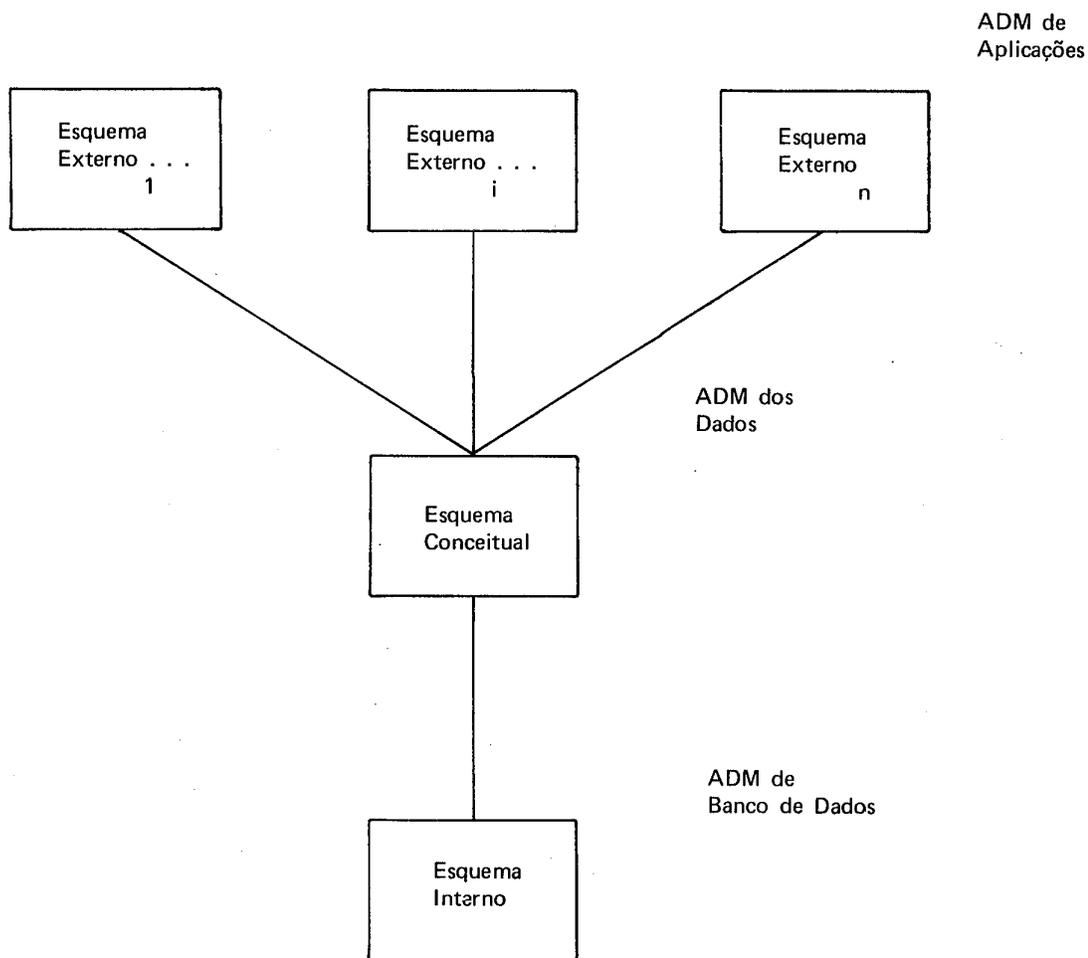


FIGURA 1 — Esquemas e Seus Administradores