

O CONCEITO DE ENTIDADE  
NA MODELAGEM SEMANTICA DE DADOS

*José Palazzo Moreira de Oliveira*

*José Mauro Volkmer de Castilho*

*Clésio Saraiva dos Santos*

Universidade Federal do Rio Grande do Sul

Porto Alegre - Brasil

Resumo

O desenvolvimento de sistemas de informação, apoiados por bancos de dados, exigiu o desenvolvimento de modelagens complexas da realidade. Os sistemas de gerência de banco de dados oferecem modelos de dados que permitem representar os objetos relevantes para uma aplicação. Os diversos modelos de dados apresentam um conjunto variado de conceitos o que leva a dificuldades na seleção das alternativas possíveis para a modelagem. Neste artigo é apresentado um modelo, o modelo E, que pelo emprego de um único conceito de base, a entidade e de operadores de tipo, permite uma representação natural e ao mesmo tempo completa da realidade modelada.

Palavras-chave

Modelo de dados, banco de dados, modelagem de sistemas de informação.

1. Introdução

A modelagem de sistemas de informação (SI) constitui-se em um dos problemas de solução mais complexa na área de processamento eletrônico de dados (PED). Isto é decorrência de ser a modelagem uma atividade interdisciplinar. É necessário o trabalho coordenado de especialistas em computação com os usuários. Os primeiros conhecem as soluções técnicas disponíveis e os segundos tem o conhecimento das necessidades das aplicações. O problema da comunicação é grave. Diversas técnicas foram desenvolvidas na área de análise de sistemas para procurar resolver a dificuldade de comunicação [Mar 78].

Em PED a modelagem dos sistemas de informação está muito ligada ao modelo de dados utilizado pelo sistema de gerência de banco de dados (SGBD) escolhido. Um dos itens mais importantes para permitir uma boa comunicação entre os usuários e os técnicos

em BD é a adequação dos conceitos existentes no modelo de dados aos objetos modelados. Neste artigo é apresentado um modelo útil na descrição de sistemas de informação e facilmente utilizável para a implantação de um SGBD.

Os diversos modelos de dados disponíveis apresentam uma grande variedade de conceitos tais como: relações, atributos, domínios [Cod 70], entidades, relacionamentos, valores [Che 76], registros, campos e "cosets" [CODASYL 71], entre outros. Uma análise apressada poderia levar à conclusão, errônea, de que quanto maior o número de conceitos disponíveis em um modelo mais facilidade o mesmo oferece para a modelagem.

Entretanto, devido à interdependência entre os conceitos pode levar o usuário a sérias dificuldades para a classificação dos objetos a serem modelados em razão da subjetividade que passa a caracterizar este processo.

Pode ser citado como exemplo o Modelo Entidade-Relacionamento (ER), que não apresenta uma perfeita distinção entre os conceitos de Entidade, Relacionamento e Valor. Em um banco de dados de ensino, turmas podem ser descritas como entidades, tendo professor, disciplina e sala como atributos, ou alternativamente, turma pode ser modelada como um relacionamento entre as entidades professor, disciplina e sala.

## 2. Os modelos semânticos

Um modelo de dados é um formalismo utilizado para representar os objetos que pertencem à parte da realidade correspondente a um grupo de aplicações. A semântica associada aos objetos corresponde às propriedades de estrutura e de comportamento dos objetos selecionados. Ela leva em consideração, em particular, o efeito dos operadores existentes sobre os objetos. Por outro lado torna-se cada vez mais importante a possibilidade de uma representação mais fiel da realidade. Esta necessidade tem sido detetada e diversos trabalhos propõem modelos ditos semânticos [Cod 79], [HM 81].

Estes modelos procuram incorporar uma maior parcela da semântica da realidade no modelo de dados através de novos tipos e de operadores de tipo. Entretanto a complexidade do modelo aumenta consideravelmente. Em particular a extensão do modelo de dados relacional [Cod 70] apresentada por Codd no modelo RM/T [Cod 79] apresenta um sem número de conceitos (entidades do núcleo, entidades características, entidades associativas, entidades do núcleo interno, "nonentity associations", generalização, agregação, "cover aggregation", generalização alternativa) que tornam seu emprego para a modelagem extremamente complexo.

A partir destas constatações, no presente artigo, é proposto

um modelo de **Entidades** (modelo **E**) baseado no conceito de entidade e nos construtores de tipo **soma**, **especialização**, **produto** e **correspondência**. O objetivo deste modelo é simplificar o processo de modelagem pelo emprego do conceito único de entidade. Desta forma é eliminada a necessidade de enquadrar um objeto em uma de várias classes alternativas. Por outro lado, os construtores de tipo proporcionam a possibilidade de combinação de entidades para a obtenção de tipos mais complexos e ajustados aos objetos que devem ser modelados. Pela simplicidade e clareza de sua definição, o modelo **E** apresenta uma definição formal simples e direta. Além disto, de um ponto de vista prático, poupa o usuário da desconfortável tarefa de descrever suas aplicações com alta dose de subjetividade e com um número de alternativas que cresce exponencialmente com o número de objetos a serem modelados.

### 3. O modelo **E**

O modelo **E** está baseado no modelo **E-R** estendido proposto em [SNF 80] como uma evolução do modelo Entidade-Relacionamento (ER) [Che 76]. O modelo **E** considera como inicial o conceito de entidade. Uma entidade corresponde a um objeto, fato ou evento sobre o qual devem ser mantidas informações.

Entidades podem compor outras entidades. A forma de caracterizar o detalhamento, ou composição, de entidades no modelo **E** é a definição de entidades por meio de construtores de tipo. Um construtor de tipo combina tipos de entidade já definidas para criar novos tipos de entidade. Como o resultado de um operador de tipo é um novo tipo de entidade as estruturas de derivação são homogêneas. Isto é, os operadores de tipo podem ser aplicados sucessivamente pois o resultado será sempre uma entidade. O modelo **E** oferece quatro construtores de tipo: **especialização**, **soma**, **produto** e **correspondência**.

#### 3.1 Especialização e Soma

Considerando o tipo de entidade **t**, uma especialização de **t** é um novo tipo de entidade **t'** cujas ocorrências são elementos de um subconjunto do conjunto de ocorrências de **t**. Por exemplo, considerando um tipo de entidade **funcionario**, o tipo de entidade **engenheiro** pode ser definido como:

**engenheiro** = especialização de (**funcionário**)

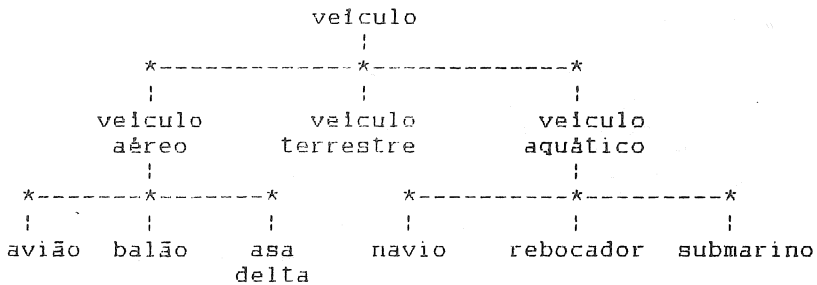
Uma observação deve ser feita aqui, em relação a este tipo de construtor. Uma entidade de tipo **engenheiro** é também uma entidade de tipo **funcionário**. Assim a criação de uma entidade do tipo **engenheiro** no banco de dados, implica na criação de uma entidade do tipo **funcionário** no mesmo banco de dados.

Se  $t_1, t_2, \dots, t_n$  são tipos de entidade, a soma de  $t_1, t_2, \dots, t_n$  é um novo tipo de entidade  $t$  cujas ocorrências são ocorrências de qualquer um dos tipos operandos. Por exemplo, considerando entidades do tipo engenheiro e técnico, o tipo de entidade membro-de-equipe pode ser definido como

membro-de-equipe = soma de (engenheiro, técnico)

O construtor soma é complementar ao construtor especialização e permite ver um tipo de objeto de nível mais alto como a composição de tipos de objetos de nível inferior.

O conceito de "soma" permite a obtenção de hierarquias de generalização. O exemplo citado por Smith e representado a seguir pode ser modelado com o auxílio deste construtor.

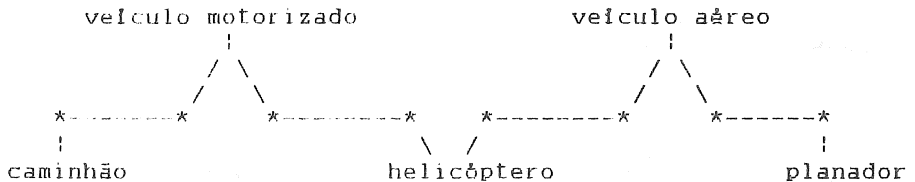


veiculo aéreo = soma de (avião, balão, asa delta)

veiculo aquático = soma de (navio, rebocador, submarino)

veiculo = soma de (veiculo aéreo, veiculo terrestre,  
veiculo aquático)

No caso em que é desejada a representação da participação de um tipo de entidade em duas hierarquias de generalização deve ser empregado o construtor de soma.



veiculo motorizado = soma de (caminhão, helicóptero)

veiculo aéreo = soma de (helicóptero, planador)

### Correspondência

Se  $t$  é um tipo de entidade, uma correspondência de  $t$  é um novo tipo de entidade  $t'$  cujas ocorrências são conjuntos de ocorrências de  $t$ . Por exemplo, um tipo de entidade equipe pode ser definida como:

equipe = correspondência de (membro-de-equipe)

A restrição de integridade associada a este construtor é:

"Uma entidade somente pode participar como componente em uma correspondência se for uma ocorrência de seu tipo."

Uma correspondência pode ser manual, quando as inserções devem ser realizadas explicitamente, ou automática. Neste último caso é especificada uma condição de pertinência que, quando satisfeita, desencadeia a inclusão automática da entidade no conjunto definido pela correspondência.

### 3.3 Produto

Os construtores soma, especialização e correspondência permitem a modelagem de interdependências entre as entidades. Esta modelagem é dependente tão somente das entidades e independe da descrição detalhada das propriedades destas entidades. Para agregar as diferentes propriedades em uma entidade é empregado o construtor produto. Este construtor é independente do grupo anterior e corresponde ao conceito de agregação de Smith & Smith.

Se  $t_1, t_2, \dots, t_n$  são tipos de entidade, um produto de  $t_1, t_2, \dots, t_n$  é um novo tipo de entidade  $t$  cujas ocorrências são  $n$ -uplas compostas com ocorrências de cada tipo participante do produto. Cada componente de uma ocorrência de  $t$  está associado ao papel que aquele componente desempenha no produto. A caracterização dos papéis existentes num produto pertence à definição do mesmo. Por exemplo, um tipo de entidade departamento pode ser definido sobre tipos funcionário, equipe e recursos como:

departamento = produto de (  
CHEFIA: funcionário,  
CORPO-TECNICO: equipe,  
INFRA-ESTRUTURA: recursos)

Os exemplos acima podem ter dado a impressão de que apenas objetos "concretos", como engenheiros, chefes, departamentos, são entidades. Pelo contrário, também objetos tais como uma determinada idade, ou um determinado salário podem ser encarados como entidades. Deste ponto de vista, pode ser definido um tipo

de entidade funcionário como o produto de tipos de entidade nome-próprio, endereço, idade, salário, como descrito a seguir:

```
funcionário = produto de (
    NOME: nome-próprio,
    RESIDENCIA: endereço,
    IDADE: idade,
    REMUNERACAO: salário )
```

A noção, geralmente difundida, de um funcionário é a de um objeto concreto ao qual estão associados valores de idade, nome-próprio, etc. Do ponto de vista da modelagem que está sendo construída, uma ocorrência de entidade **funcionario** é apenas uma ocorrência do produto definido, e não o **funcionario** (pessoa) em si. A associação entre um funcionário (pessoa) e a entidade do tipo **funcionario** correspondente faz parte do processo de abstração executado pelo analista durante a definição do modelo da aplicação.

Com os conceitos de entidade e de operadores de tipo é possível modelar um conjunto complexo de objetos do mundo real por meio de tipos definidos de forma natural, isto é, próxima à estrutura percebida na realidade. Para isto é de grande valia a possibilidade de combinação sucessiva de operadores de tipo. A decisão fica restrita à escolha da combinação de operadores de tipo necessários para representar uma dada percepção da realidade.

O anexo 1 apresenta definições de tipos de entidades para uma aplicação relativa a projetos e departamentos.

#### 4. Base de dados

Uma tipo **base de dados**, no modelo E, é também um tipo de entidade, mais precisamente, de uma correspondência definida sobre os tipos de entidade identificados no processo de modelagem de dados. Por exemplo, parte da definição de uma base de dados envolvendo os tipos de entidade apresentados como exemplo na seção anterior pode ser descrita como abaixo:

```
base-de-dados = correspondência de (
    soma de (
        NOMES-PROPRIOS;
        SALARIOS;
        TAREFAS;
        FUNCIONARIO;
        DEPARTAMENTOS;
    )
).
```

A ocorrência do tipo de entidade base-de-dados, em um certo instante, é vista como o estado da base de dados naquele instante. Não está excluída a possibilidade de coexistirem várias ocorrências da base de dados, correspondendo a vários estados da base em diferentes instantes.

A interdependência entre ocorrências de tipos de entidades diferentes na base de dados é função da interdependência entre os tipos correspondentes. A associação fica estabelecida implicitamente pelo mecanismo que garante a restrição de integridade estrutural existente entre tipos e suas especializações. Assim a existência de uma entidade de tipo **engenheiro** na base de dados implica na existência de uma correspondente entidade de tipo **funcionario**.

## 5. Mecanismo de construção e referência a entidades

As entidades que participam de uma ocorrência ou estado da base de dados podem ser interdependentes, como mencionado anteriormente. A estrutura geral das interdependências é vista como o esquema de caminhos de acesso às entidades. Assim, por exemplo, uma entidade de tipo **funcionario** pode estar relacionada a uma entidade de tipo **engenheiro** (pois **engenheiro** é uma especialização de **funcionario**) e também estar relacionada com uma entidade de tipo **departamento** (é o componente com papel de CHEFIA), e indiretamente relacionada com uma entidade de tipo **projeto** (é componente com papel de LIDER no projeto).

O esquema de interdependência é montado com a definição de "operadores de referência", que apresentam ou denotam a entidade relacionada com a entidade dada. Por exemplo, seja **n** uma entidade do tipo nome-próprio. A entidade do tipo **engenheiro** relacionada com **n** seria denotada por:

**engenheiro [ NOME = n ]**

que se lê como "a entidade de tipo **engenheiro**, onde **n** é componente com papel NOME".

### 5.1 Construção de ocorrências de entidades

Os operadores **Cria**, **Destroi**, **Altera**, **Insera** e **Remove** permitem a manipulação das ocorrências na base.

**Cria (t, e)**

cria uma ocorrência da entidade **t** e corpo **e**. O corpo é uma apresentação textual da entidade montada de forma conveniente.

**Destroi (t, e)**

destroi uma ocorrência e de entidade do tipo  $t$ .

**Altera ( $t, e, p, el$ )**

na ocorrência e de entidade do tipo  $t$ , definido como um produto,  $el$  passa a ser o componente com papel  $p$ .

**Insere ( $t, e, el$ )**

na ocorrência e da entidade do tipo  $t$ , definido como uma correspondência manual, insere a ocorrência de entidade  $el$  como elemento.

**Remove ( $t, e, el$ )**

na ocorrência e de entidade do tipo  $t$ , definido como uma correspondência manual, remove a ocorrência de entidade  $el$  do conjunto.

### 5.2 Operadores de referência a entidades

Considerando  $e$  uma entidade,  $t$  um tipo de entidade e  $p$  um papel, definimos, informalmente os cinco operadores que se seguem:

**Existe ( $t, e$ )**

tem o valor VERDADEIRO se existe na base de dados uma ocorrência e de entidade do tipo  $t$ .

**$t [p = e]$  (operador de seleção)**

denota a entidade de tipo  $t$  da qual  $e$  é o componente com papel  $p$ .

**componente ( $t, e, p$ )**

denota a entidade que é o componente da ocorrência  $e$ , do tipo  $t$ , com papel  $p$ .

**conjunto ( $t, e$ )**

denota a entidade do tipo  $t$  que é uma correspondência da qual  $e$  é membro.

**elemento ( $t, e$ )**

denota uma entidade que é um elemento da entidade  $e$  do tipo  $t$ , definido como uma correspondência.

### 5.3 Exemplos de referências a entidades

A seguir são apresentados alguns exemplos de aplicação dos



operadores de referência a entidades.

a) turma [código = "T500"]

denota a ocorrência de entidade do tipo turma que tem "T500" como componente com papel código.

b) componente (turma, turma [código = "T500"], professor)

denota a ocorrência da entidade que é componente com o papel professor da turma "T500".

c) conjunto (alunos, aluno [nome = "André"])

denota a entidade do tipo alunos (correspondência) que tem o aluno André como um de seus elementos.

d) elemento (alunos, correspondência (turma, turma [código = "T500"], alunos)

denota um dos elementos (aluno) do conjunto de alunos da turma "T500".

Cada operador leva à denotação de de apenas uma entidade. Caso haja mais de uma candidata à denotação, uma delas será escolhida arbitrariamente. O operador `próx` aplicado aqueles operadores (seleção, conjunto e elemento) que comportam a existência de várias entidades candidatas, denota uma das candidatas ainda não excolhida desde a última aplicação do operador de referência.

```
e)  n1 <-- cria (nome_próprio, "João")
     n2 <-- cria (nome_próprio, "Pedro" )
     n3 <-- cria (nome_próprio, "Paulo" )
     n  <-- cria (nomes, n1, n2, n3)
     n4 <-- elemento (nomes, n)
     n5 <-- próx (nomes, n)
     n6 <-- próx (nomes, n)
```

Neste exemplo, `n4`, `n5`, `n6` denotam as três entidades do tipo `nome_próprio` que integram o conjunto `n`.

## 6. Conclusão e trabalhos futuros

Ao longo deste artigo foi descrito um modelo de dados que, utilizando o conceito de tipo de dados e construtores, permite uma modelagem da realidade mais simples. As decisões do analista de Sistema de Informações ficam facilitadas por não ter que tomar decisões subjetivas sobre a classificação dos objetos da realidade em diferentes classes parcialmente sobrepostas de conceitos no modelo.

O trabalho de pesquisa está se desenvolvendo em diversas frentes: especificação formal do modelo de dados, desenvolvimento de uma metodologia de projeto de esquema de banco de dados utilizando os conceitos do Modelo E e estudos para implementação de um SGBD.

### Bibliografia

- [Che 76] PPS Chen, "The entity relationship model: toward a unified view of data.", ACM TODS, v 1, n 1, pag 9, 1976.
- [Cod 70] EF Codd, "A relational model of data for large shared data banks.", Communications of ACM, v 13, n 6, pag 377, 1970.
- [Cod 79] EF Codd, "Extending the database relational model to capture more meaning.", ACM TODS, v 4, n 4, 1979.
- [CODASYL 71] Data Base Task Group of CODASYL Programming Language Committee. Report, apr 71.
- [LPV 83] M. Lopez, J. Palazzo Oliveira & F. Velez, "the TIGRE Data Model.", relatório de pesquisa Tigre n 2, IMAG, Grenoble, França, nov 83.
- [HM 81] M Hammer & D McLeod, "Database description with SMD: a semantic database model.", ACM TODS, v 6, n 3, pag 351, 1981.
- [Mar 78] T. DeMarco, "Structured analysis and system specification.", New York, Yourdon Inc., 1978.
- [SNF 80] CS Santos, EJ Neuhold & AL Furtado, "A data-type approach to the entity-relationship model.", in Entity-relationship approach to systems analysis and design, ed PPS Chen, North Holland, 1980.
- [SS 77] JM Smith & DCP Smith, "Database abstractions: aggregation and generalization.", ACM TODS, v 2, n 2, pag 105, 1977.

Apêndice

## Definição de tipos de entidade

nome\_próprio.

endereço.

salário.

idade.

equipamento.

tarefa.

funcionário = produto de (  
    NOME: nome-próprio,  
    RESIDENCIA: endereço,  
    IDADE: idade,  
    REMUNERACAO: salário).

engenheiro = especialização de (funcionário).  
    FORMACAO: ramo-de-engenharia,  
    DATA-FORMATURA: data).

técnico = especialização de (funcionário).  
    ESPECIALIDADE: especialidade-técnica).

máquina = especialização de (equipamento).

membro-de-equipe = soma de (engenheiro, técnico).

equipe = correspondência de (membro-de-equipe).

datilógrafo = especialização de (funcionário).

departamento = produto de (  
    CHEFIA: funcionário,  
    CORPO-TECNICO: equipe,  
    INFRA-ESTRUTURA: correspondência de (equipamento)).

projeto = produto de (  
    LIDER: engenheiro,  
    PLANO: correspondência de (tarefa),  
    EQUIPE: equipe,  
    RECURSOS: correspondência de (máquina)).