

Evolução de Esquemas Conceituais: o Conceito de Papel

**Nina Edelweiss
José Palazzo M. de Oliveira
Graziela Kunde**

Universidade Federal do Rio Grande do Sul
Curso de Pós-graduação em Ciência da Computação
Av. Bento Gonçalves, 9500 - Bloco IV - Agronomia - Caixa Postal 15.064
CEP 91501-970 - Porto Alegre - RS - Brazil
Fax +55(51)336.5576
e-mail: [palazzo, nina] @inf.ufrgs.br

Resumo

O esquema conceitual de um banco de dados pode ser modificado durante sua existência devido a modificações ocorridas nos requisitos das aplicações. Estas modificações podem ocorrer devido a diferentes circunstâncias, tais como mudanças nas legislações ou nas regras de trabalho da organização. Alterações dos requisitos implicam em modificações tanto no esquema conceitual quanto na representação dos dados armazenados, para adaptá-los ao novo esquema. O objetivo deste artigo é de apresentar uma alternativa para suportar a evolução no comportamento dos objetos representados, tanto a nível de dados armazenados, como a nível de esquema conceitual. Esta alternativa é viabilizada pelo conceito de papel em um banco de dados temporal orientado a objetos.

Abstract

During a database system's lifetime the conceptual schema may be modified due to the application requirements' evolution. These modifications occur due several different circumstances, like changes of legislation or in the organization's rules. Due to these modifications not only the conceptual schema shall change, but also the already stored data shall be adapted to the new schema. This paper presents an approach to support objects' behavior evolution, not only considering stored data but also the conceptual schema. The present approach is based on the use of the role concept associated to a temporal object-oriented database.

Palavras chave: Banco de dados, Esquema conceitual, Orientação a objetos

1. Introdução

Os atuais sistemas de informação apresentam requisitos muito complexos que necessitam de características de suporte evoluídas dos SGBDs. Entre estas características encontram-se as ligadas à evolução do esquema conceitual e a conseqüente modificação na extensão do BD. O esquema conceitual de um banco de dados representa os requisitos das aplicações suportadas na forma de estrutura de dados, restrições de integridade e, nos modelos mais evoluídos, regras de transição de estados. Como os dados evoluem ao longo do tempo é necessária a inclusão desta variabilidade nos modelos de dados. Esta necessidade levou ao desenvolvimento dos bancos de dados temporais [Clifford 95, Tansel 93]. Entretanto, os requisitos e as restrições do sistema de informações, suportado pelo SGBD, também evoluem ao longo do tempo implicando em evolução do esquema conceitual. Para permitir uma representação precisa e completa de uma parte relevante do mundo real é necessário que a evolução do esquema conceitual seja suportada pelo sistema de gerência do banco de dados.

A evolução do esquema pode ser representada pela criação de uma nova versão do mesmo. A definição de uma nova versão afeta os dados existentes no banco de dados. Nos sistemas tradicionais de bancos de dados, a cada modificação no esquema conceitual corresponde uma adaptação, realizada como um procedimento administrativo de transformação, no conjunto de dados afetado pela modificação do esquema. Esta dependência entre o esquema e os dados leva a uma série de problemas na transformação dos dados, com a eventual perda de informações históricas. Alternativamente podem ser mantidas versões sucessivas dos estados dos esquemas, cada uma associada com um subconjunto de dados. Neste caso são possíveis consultas sobre um estado passado da extensão do banco de dados associado à respectiva versão do esquema.

Vários estudos sobre a evolução de esquemas foram desenvolvidos considerando modelos relacionais, entidade-relacionamento e orientados a objetos [Castilho 93, Ewald 93, Lerner 90, Penney 87, Silva 94, Zicari 91]. Entretanto, a maioria destes estudos não trata de bancos de dados temporais - somente uma versão do esquema pode existir a cada instante, devendo toda extensão do banco de dados ser adaptada para esta versão. Recentemente tem sido analisada a possibilidade de tratar da evolução de esquemas quando utilizados bancos de dados temporais na extensão do BD [Kim 95, McKenzie 90, Roddick 92] e também para armazenar as diferentes versões de esquemas [DeCastro 95, Edelweiss 95].

Nos modelos de dados convencionais orientados a objetos, um objeto é criado como uma instância de uma classe e apresenta um identificador único que o diferencia dos outros objetos. Um objeto sempre pertence à mesma classe durante todo o seu tempo de vida, até mesmo se suas características comportamentais mudarem. Em alguns modelos é permitida alguma forma limitada de troca de classe - é possível a um objeto migrar entre uma classe e sua subclasse ou entre diferentes classes [Tsichritzis 87]. A necessidade de um objeto mudar dinamicamente de classe é muito comum, como fica evidenciado através do seguinte exemplo: uma pessoa pode passar de estudante de graduação para estudante de pós-graduação e, a seguir, para pesquisador associado. Além disso, é necessária a possibilidade de modelar uma mesma pessoa segundo percepções diferentes. No exemplo anterior poderia ser necessário manter o estado contratual: bolsista do CNPq, bolsista de uma empresa, recém doutor, contrato por tempo determinado. Para garantir a classificação exclusiva dos objetos é necessária a geração de uma classe para cada combinação possível de percepções, o que constitui uma solução muito pouco natural.

Para permitir uma representação mais natural da realidade são necessárias extensões dos modelos tradicionais. O conceito de *papel* [Pernici 90] estende os modelos orientados a objetos tradicionais, possibilitando a representação dos diferentes comportamentos como parte de um mesmo objeto e permitindo a modelagem natural da evolução dinâmica do objeto.

O objetivo deste artigo consiste em mostrar a utilidade do conceito de papel para suportar a evolução de esquemas de bancos de dados com um impacto mínimo na modificação do esquema e na adaptação dos dados já armazenados.

O artigo está organizado da forma a seguir. A seção 2 apresenta o conceito de papel. Um modelo orientado a objetos que utiliza este conceito é apresentado na seção 3. A seção 4 analisa os diferentes bancos de dados com relação a informações temporais que podem ser utilizadas para representar a extensão e os esquemas, quando estes são versionados. A evolução de esquemas quando utilizado um modelo orientado a objetos com papéis é analisada na seção 5. Para exemplificar as idéias expostas, na seção 6 é apresentada uma forma de implementar o modelo orientado a objetos com papéis em um banco de dados relacional, destacando-se possíveis evoluções do esquema e como estas se refletem nos dados armazenados.

2. O Conceito de Papel

Para representar os diferentes comportamentos de um objeto pode ser utilizado o conceito de papel. Este conceito, definido dentro do objeto, permite a encapsulação da representação dos diferentes comportamentos assumidos por uma entidade do mundo real. Uma classe de objetos pode apresentar diferentes papéis, os quais podem ser desempenhados por cada uma de suas instâncias. Um objeto continua sendo instância de uma só classe, mas desempenha diferentes papéis durante sua existência. O conceito de papel permite a representação da evolução do comportamento de um objeto, uma vez que a instanciação de papéis é interna ao objeto. Os papéis são instanciados dinamicamente - instâncias de papéis de um objeto são criadas e destruídas livremente durante a sua existência. Além disso, um objeto pode desempenhar simultaneamente mais de um papel, e pode apresentar mais de uma instância de um mesmo papel no mesmo instante de tempo. A abstração de generalização/especialização continua existindo, estendendo-se o conceito de herança também para os papéis.

A figura 1 apresenta um exemplo de classes/subclasses com papéis. É representada uma classe *pessoa*, a qual pode apresentar dois comportamentos distintos - *aluno* e *funcionário*. Um objeto da classe *pessoa* que fosse somente estudante apresentaria uma instância do papel *aluno*. A evolução do objeto trocando de papel pode ser representada através de criação de uma instância de outro papel e da destruição da instância existente. O fato de um estudante ser também funcionário, dois papéis desempenhados simultaneamente, seria representado por uma instância de *estudante* e uma instância de *funcionário*, as duas coexistindo. O fato de uma mesma pessoa apresentar mais de um emprego pode ser representado por duas ocorrências simultâneas do mesmo papel (*funcionário*). No exemplo, a classe *pessoa* apresenta duas subclasses: *homem* e *mulher*. Em cada uma das subclasses os papéis *aluno* e *funcionário* estão presentes. As subclasses especializam propriedades e/ou métodos da superclasse, tanto a nível de classe como um todo, como a nível de papéis.

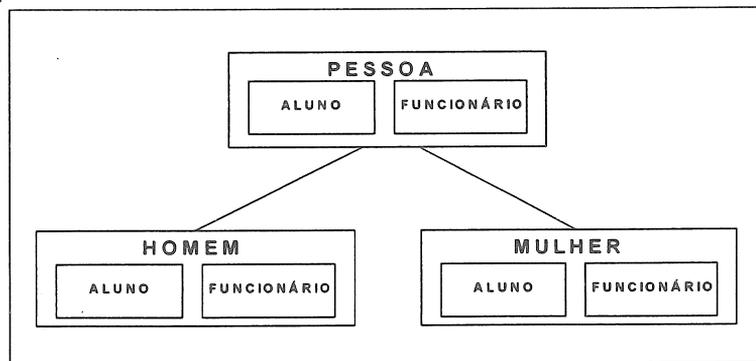


Figura 1: Modelo orientado a objetos com papel

Para explicar melhor o conceito de papel pode ser apresentado o seguinte caso [Wieringa 91]:

“Assuma que passageiro é uma subclasse de pessoa e considere uma pessoa que migra para a subclasse passageiro, entrando num ônibus. Este ônibus pode carregar 4000 passageiros em uma semana, mas contando diferentemente, ele pode carregar 1000 pessoas na mesma semana. Então contar pessoas difere de contar passageiros.”

Se *passageiro* for modelado como uma subclasse de *pessoa*, então contar passageiros seria o mesmo que contar pessoas. Com isso, o identificador de *passageiro* é também um identificador de *pessoa*. Como passageiro não é idêntico a uma pessoa aparentemente *pessoa* e *passageiro* têm

identificadores diferentes. Devem existir, também, formas diferentes para representar esta instâncias. Neste exemplo é evidente que *passageiro* não é idêntico a *pessoa* mas sim um estado de *pessoa*, ou mais adequadamente um papel desempenhado por uma pessoa. Então contar passageiros significa contar quantas pessoas desempenharam o estado de ser um passageiro.

Em uma primeira análise poderíamos ser induzidos a pensar que os papéis representam apenas estados particulares que uma entidade pode assumir durante sua vida e, assim, serem implementados como um atributos temporais multi-valorados. Isto não é possível, em geral, pois um objeto, desempenhando um papel, tem um comportamento específico àquele papel que é especificado na definição do papel, coisa que não pode ser representada na forma tradicional de modelar classes. No caso de uma modelagem utilizando uma especialização parcial estaríamos indicando que é possível a existência de pessoas que não sejam passageiros. Mas neste caso o fato de uma pessoa ser, repetidas vezes, passageiro não é adequadamente representado. Como exemplo limite, o caso de uma pessoa que é passageiro de um carro e, ao mesmo tempo, passageiro de um ferry (o carro embarcou) não é adequadamente representado pela modelagem de especialização.

Neste exemplo fica clara a representação de um comportamento dinâmico de um objeto no banco de dados através de papéis. É evidente que os diferentes papéis desempenhados por um objeto devem estar representados, necessariamente, no esquema conceitual. Isto feito, a migração do objeto entre diferentes papéis ou a participação múltipla em um mesmo papel é facilmente realizável.

3. Modelo de Dados Orientado a Objetos com Papéis

O principal conceito de modelagem de um modelo de dados orientado a objetos com papéis é o “objeto”. Este conceito continua sendo fundamental nos modelos orientados a objetos que utilizam o conceito de papel. Como em todo modelo orientado a objetos, os objetos dos modelos que utilizam papéis apresentam propriedades (atributos) e métodos, são instâncias de classes e evoluem com o passar do tempo. Toda classe é identificada por um nome único. Dois tipos de propriedades podem ser identificados quando considerados modelos temporais: *propriedades estáticas*, que apresentam sempre o mesmo valor, e *propriedades dinâmicas*, que podem ter seu valor alterado com o passar do tempo. Os modelos apresentam, ainda, alguma forma de definir regras de integridade estáticas e dinâmicas.

O conceito de papel permite que diferentes papéis possam ser definidos em uma classe. Cada papel deverá ter um nome único na classe, podendo apresentar propriedades e métodos próprios. As propriedades e métodos definidos dentro da classe se aplicam a todos os papéis. Além disso, a classe apresenta métodos que manipulam a criação, destruição, suspensão temporária e retomada dos papéis. Estas propriedades e métodos da classe são geralmente definidos em um papel especial, no qual são também definidas as propriedades comuns a todos os papéis.

O modelo TF-ORM (*Temporal Functionality in Objects with Roles Model* [Edelweiss 93a,b]), utilizado nos exemplos deste artigo, utiliza esta estruturação geral. Neste modelo uma classe e cada um de seus papéis são definidos através de um nome, um conjunto de propriedades que o objeto apresenta neste papel, um conjunto de estados abstratos que o objeto pode assumir neste papel, um conjunto de mensagens que podem ser enviadas e recebidas por este papel e um conjunto de regras (regras de transição de estados e regras de integridade). Fórmulas lógicas

podem ser associadas às regras de transição, atuando como restrições às possíveis transições de estados.

A evolução temporal dos objetos do modelo TF-ORM, tanto a nível de existência do objeto e de seus papéis quanto a nível de valores de propriedades, é registrada no banco de dados que implemente o modelo. A evolução temporal dos objetos e de seus papéis é registrada em propriedades pré-definidas. Já a evolução temporal dos valores das propriedades é identificada por rótulos temporais (tempo de transação e de validade) associados a cada valor definido. Os identificadores de instância da classe e de instâncias de papéis também são armazenados em propriedades pré-definidas - *old* e *rid*, permitindo a identificação da evolução de cada papel de cada objeto claramente. As propriedades citadas podem ser consultadas pela linguagem de consulta do modelo [Edelweiss 94], permitindo, desta forma, a resolução de consultas relativas ao estado atual da base de dados e a todos os seus estados passados e futuros.

4. Evolução do Esquema Conceitual

Os *bancos de dados não temporais* possuem um esquema estático e uma correspondente extensão estática. Mudanças na definição do esquema ou no conteúdo do banco de dados não deixam traço e não é possível realizar consultas sobre dados antigos. *Bancos de dados temporais convencionais* podem ser vistos como possuindo um esquema conceitual estático e extensões dinâmicas, incorporando a história dos dados armazenados. Uma extensão dinâmica corresponde a uma seqüência de estados estáticos (versões) desta extensão.

Em [Edelweiss 95] foi apresentado o conceito de *banco de dados temporal generalizado* - um banco de dados cujos membros são sistemas de bancos de dados temporais convencionais, cada um apresentado um esquema conceitual e um conjunto de seqüências de extensões do banco de dados. Este tipo de banco de dados permite representação temporal tanto na intenção (evolução dos requisitos da aplicação, representada pelas diversas versões do esquema conceitual) como na extensão (evolução dos valores armazenados) do banco de dados.

Bancos de dados temporais apresentam rótulos temporais associados aos dados. Na classificação proposta em [Snodgrass 85] e refinada em [Tansel 93] são identificadas as seguintes categorias de bancos de dados, de acordo com o(s) tipo(s) de rótulo(s) temporal(is) utilizado(s): (i) *bancos de dados instantâneos*, os bancos de dados comuns, não temporais, que não apresentam informações temporais associadas aos dados; (ii) *bancos de dados de tempo de transação*, onde somente o tempo de transação é associado aos dados, correspondendo este, portanto, à validade dos dados; (iii) *bancos de dados de tempo de validade*, que utilizam o tempo de validade da informação e não o tempo em que a informação é introduzida no banco de dados; e (iv) *bancos de dados bitemporais*, os quais apresentam os dois tempos - o tempo de transação e o tempo de validade.

As diversas versões de um esquema também são armazenadas em um (meta)banco de dados. Diversas formas de armazenamento podem ser utilizadas para armazenar estas versões, dependendo das informações temporais associadas a cada uma delas. A seguinte classificação foi proposta em [Edelweiss 95]: (i) *esquemas instantâneos*, quando somente uma versão de esquema está disponível; (ii) *esquemas de tempo de transação*, quando a cada alteração do esquema que gere uma nova versão, é associado o tempo em que a alteração foi registrada no banco de dados; (iii) *esquemas de tempo de validade*, quando cada alteração é associada ao tempo em que inicia a validade da alteração; e (iv) *esquemas bi-temporais*, quando as alterações são associadas com ambos os tempos de transação e de validade.

Adotamos, neste artigo, o caso em que o esquema é representado por um banco de dados de tempo de transação, sendo portanto as versões do esquema associadas ao seu tempo de transação. A partir deste pressuposto, vamos derivar as conseqüências benéficas da utilização do conceito de papel para assegurar a consistência entre o esquema conceitual e a extensão do banco de dados.

Quando a evolução do esquema conceitual é representada através de um banco de dados de tempo de transação, as sucessivas versões do esquema conceitual são acessíveis, cada uma delas associada com o correspondente tempo de validade. Transformações elementares no esquema geram uma nova versão do esquema. Uma nova versão do esquema se torna válida no momento de sua definição, assim permanecendo até que nova versão seja definida. A cada vez que uma modificação no esquema for efetuada devem ser feitas as necessárias adaptações na extensão do banco de dados, relativas à existência de objetos, papéis e propriedades afetados pela modificação.

5. Evolução de Esquemas em um Modelo OO com Papéis

A utilização de papéis facilita a definição de novas versões de um esquema por particionar os diferentes comportamentos de um objeto em representações separadas. Nos modelos que não utilizam papéis, uma alteração no comportamento do objeto (alteração em propriedades e/ou métodos) implica em um estudo de toda a estrutura deste objeto, para verificar onde esta alteração de comportamento se reflete. Nos modelos que utilizam papéis para modelar os comportamentos, a alteração será efetuada somente dentro de um determinado papel, diminuindo o escopo de alcance e a necessidade de adaptação do restante. É somente necessário analisar o papel cujo comportamento sofreu alteração, e os papéis que se comunicam com ele, de modo a adaptá-los ao novo comportamento.

A adaptação da extensão decorrente de uma modificação no esquema também é simplificada com a utilização de papéis. Somente precisam ser analisados (e, possivelmente, adaptados) os valores armazenados correspondentes a propriedades e estados do papel modificado.

Atualmente está sendo desenvolvido um estudo exaustivo da semântica das transformações de esquemas OO e das possibilidades de representação destas modificações com o conceito de papel.

5.1. Definição de Novas Versões de um Esquema

Considerando que o (meta)banco de dados que armazena as diversas versões do esquema é do tipo tempo de transação, uma nova versão de um esquema é definida quando um determinado conjunto de alterações é inserido no banco de dados. Para o modelo orientado a objetos com papéis descrito na seção anterior, um novo esquema pode ser definido através de alterações (i) a nível de classes, e (ii) a nível de papéis. As possíveis *alterações a nível de classes* são:

- definição de uma nova classe;
- retirada de uma classe;
- modificações de informações relativas a uma classe.

As vantagens da utilização de papéis aparecem nas alterações de informações relativas a uma classe. Nestas, além da possibilidade de alterações em atributos e métodos comuns a todos

os papéis, é possível separar as alterações relativas a comportamentos particulares do objeto (papéis). As seguintes operações podem ser identificadas como *alterações a nível de papéis*:

- inclusão de um novo papel;
- retirada de um papel;
- alteração de informações relativas a um papel.

As operações de inclusão e retirada de um papel são as mais comuns quando utilizados modelos de dados deste tipo. Traduzem os casos reais em que novos comportamentos são definidos para um objeto, ou que um determinado comportamento não será mais suportado. São muito simplificadas pelo contexto a ser analisado - somente devem ser considerados os controles da classe relativos ao papel considerado e às comunicações deste papel (troca de mensagens) com outros papéis. As únicas alterações que podem ser necessárias nos demais papéis desta ou de outras classes se refere aos mecanismos de comunicação destes com o papel considerado. Estas modificações não alteram os dados já armazenados no banco de dados.

As operações de alteração de informações relativas a um papel podem ser mais amplas:

- alterações em propriedades (definição de novas propriedades, remoção de propriedades, modificação do domínio de propriedades);
- alteração em métodos;
- alterações de interfaces de comunicação entre classes/papéis.

Mesmo nestes casos, a diminuição do contexto a ser analisado diminui o domínio de influência das alterações efetuadas, facilitando a adaptação das demais classes/papéis no novo esquema.

5.2. Adaptação da Extensão

Cada vez que uma nova versão do esquema é definida, informações armazenadas na extensão a partir deste momento serão definidas por esta versão. Entretanto, as informações anteriormente armazenadas continuam disponíveis, devendo ser manipuladas seguindo a definição da versão do esquema vigente no momento referenciado. Existe, portanto, uma dependência temporal entre informações armazenadas na extensão e as versões do esquema conceitual.

A definição de uma nova versão do esquema não significa que as informações anteriormente definidas na extensão percam sua validade - deve ser efetuada uma adaptação da extensão, apropriando as informações existentes à nova versão do esquema.

Das possíveis modificações no esquema, as mais comuns, quando utilizado o conceito de papel, e as consequentes adaptações a serem feitas na extensão são as seguintes:

- definição de um novo papel - devem ser inseridas na extensão as propriedades estáticas deste papel, inicialmente com valor *null*;
- remoção de um papel existente - todas as informações relativas a este papel, que estejam válidas no momento em que o novo esquema é definido, devem ter seu tempo de validade final ajustado para este mesmo instante, indicando que sua validade termina no momento em que o novo esquema é definido;
- alterações em um determinado papel:
 - ⇒ em propriedades - novas propriedades definidas devem ser inicializadas, propriedades removidas devem ter seu tempo de validade finalizado;

⇒ em métodos e interfaces - não se refletem na extensão do banco de dados.

Quando utilizado um modelo que associe às informações somente o tempo de validade inicial, nenhuma informação adicional precisará ser adicionada - estas informações somente serão recuperadas através dos esquemas associados, indicando a nova versão do esquema que deixaram de ser válidas. A linguagem de recuperação de informações deverá fazer a consistência entre os esquemas, recuperando cada informação armazenada através de seu esquema correspondente.

5.3. Exemplo de Evolução de Esquema OO com Papéis

O exemplo a seguir mostra claramente como se efetua a evolução de esquemas ao nível de papéis. Considere uma empresa composta por departamentos e pessoas empregadas pela empresa - figura 2, ponto T_A do esquema. A empresa mantém um programa de bolsas-estágio garantindo a possibilidade de estudantes de graduação desenvolverem suas habilidades realizando atividades práticas ligadas à produção. Há um funcionário associado a cada bolsista desempenhando o papel de *tutor*. Uma pessoa dentro desta empresa pode apresentar, portanto, três papéis distintos: funcionário, bolsista e tutor.

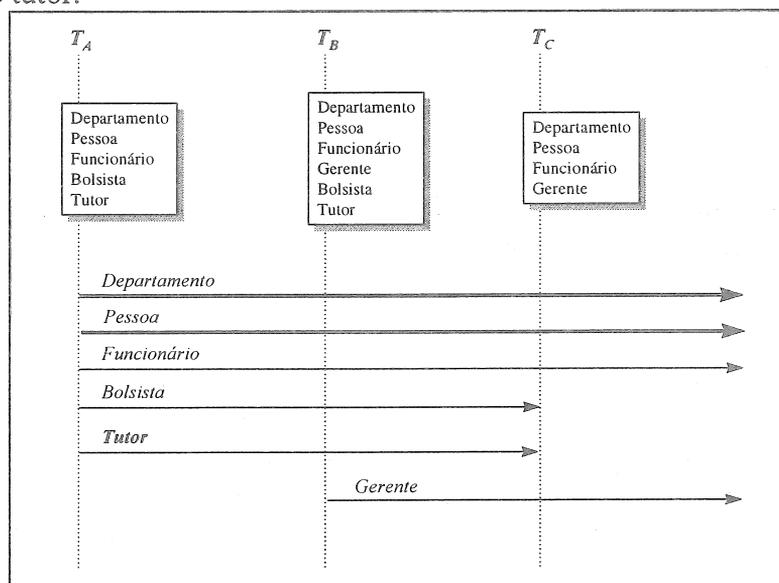


Figura 2: Evolução do Esquema Conceitual

Com o passar do tempo esta empresa evolui - figura 2, ponto T_B do esquema - e torna-se necessário incluir informações sobre os gerentes dos departamentos. Cada departamento tem um gerente, que deve ser um dos funcionários. Surge assim mais um possível papel para uma pessoa na empresa - o papel de *gerente*.

Posteriormente, devido a restrições orçamentárias, a companhia decidiu reduzir as suas despesas e eliminar o programa de bolsistas pois o custo associado à manutenção dos tutores não compensava, na opinião da direção, os lucros obtidos pela produtividade dos bolsistas. Assim, figura 2, ponto T_C do esquema - os papéis *tutor* e *bolsista* foram eliminados.

6. Implementação do Modelo OO com Papéis sobre um Banco de Dados Relacional, considerando Evolução de Esquemas

A utilização de um modelo de dados orientado a objetos com papéis, com grande poder de expressão, não implica que seja necessária a existência de um banco de dados correspondente a este modelo. A principal característica deste modelo pode ser considerada a sua grande

expressividade sendo, portanto, uma poderosa ferramenta para a modelagem conceitual de sistemas de informação. Entretanto, para a sua utilização em ambientes de produção seria necessária a existência de um SGBD comercial com todas as características de robustez necessárias a estas aplicações. Para a utilização destes modelos de dados evoluídos na modelagem de sistemas de informação faz-se necessária a conversão (mapeamento) dos conceitos desenvolvidos para estruturas de SGBDs existentes, essencialmente baseados no modelo relacional. A medida em que estes SGBDs forem incorporando características orientadas a objetos os conceitos dos dois níveis ficarão mais próximos facilitando o mapeamento.

Diversos trabalhos têm sido realizadas mapeando modelos ricos em expressividade para bancos de dados comerciais, inclusive apresentando outras formas de representação conceitual [Oliveira 95]. Como exemplo deste enfoque, apresentamos a seguir um possível mapeamento do modelo TF-ORM para um banco de dados relacional, destacando como seria tratada, neste caso, a evolução do esquema conceitual.

Examinemos a representação exemplo apresentado na seção anterior no SGBD relacional. O SGBD utilizado para esta implementação deve permitir a criação dinâmica de tabelas relacionais. Para representar o esquema conceitual é criada uma relação, correspondente ao meta-esquema, contendo as tabelas existentes na implementação. O processo de modificação do esquema conceitual corresponde à inserção da descrição do papel no meta-esquema e da criação das correspondentes tabelas no banco de dados. O meta-esquema é utilizado para manter a informação sobre os períodos de existência dos objetos e papéis no esquema conceitual. A seguir é apresentada uma versão simplificada do mecanismo.

Tabela 1: Meta-esquema

M_oid	tipo	data criação	data fim	Nome da tabela
M ₁	ob.	T _A		Pessoa
M ₂	ob.	T _A		Departamento
M ₃	pap.	T _A		Funcionário
M ₄	pap.	T _B		Gerente
M ₅	pap.	T _A	T _C	Bolsista
M ₆	pap.	T _A	T _C	Tutor

No estudo de caso encontramos as entidades *pessoa* e *departamento* representadas por uma linha dupla (\Rightarrow) na figura 2. No meta-esquema, tabela 1, estes objetos são descritos pelas tuplas com M_oid M₁ e M₂. Os papéis definidos no instante T_A, *funcionário*, *bolsista* e *tutor*, são descritos pelas tuplas M₃, M₅ e M₆, tendo cada um o tempo de transação igual a T_A. O papel de *gerente*, criado posteriormente, é representado pela tupla M₄, com tempo de transação igual a T_B. Como os papéis *bolsista* e *tutor* deixam de existir em T_C é registrada a data de finalização como T_C. Caso o papel fosse suspenso, criando períodos sucessivos de atividade e inatividade, isto seria registrado como uma entrada nesta tabela para cada intervalo de atividade.

Na extensão do banco de dados os objetos são modelados como relações que possuem um *surrogate oid*. Um papel é representado por uma tabela independente com um atributo *oid* – chave estrangeira – referenciando a ocorrência do papel básico do objeto. No estudo de caso isto é representado pelas tabelas *funcionário* e *tutor*, que representam os papéis desempenhados por *pessoa*.

O valor 4, chave estrangeira nas tuplas das tabelas *funcionário* e *tutor* identifica-os como papéis desempenhados pela mesma pessoa.

Tabela 2: Pessoa

oid	estado	gênero	data nascimento
3	ativo	M	15/12/56
4	ativo	F	09/07/60
5	ativo	M	06/06/51
6	ativo	M	21/05/70

Tabela 3: Funcionário

oid	rld	val_id	estado	data transação	data fim
3	6	6	ativo	10/03/90	
4	8	8	ativo	15/03/90	
5	2	2	ativo	05/04/90	
6	7	7	ativo	10/04/90	

Tabela 4: Tutor

oid	rld	val_id	estado	data transação	data fim
4	1	1	terminado	15/03/90	T _c
5	2	2	terminado	01/05/90	T _c
7	3	3	terminado	03/04/90	T _c

As propriedades dinâmicas, ou seja, aquelas que mudam ao longo do tempo, são representadas de maneira diferente das estáticas. Para o nome de pessoa, por exemplo, que é uma propriedade dinâmica, deve ser construída uma tabela separada da tabela *pessoa*. A cada novo nome para uma pessoa, é colocada uma nova tupla nesta tabela. Assim como em *funcionário* e *gerente*, o identificador *oid* é repetido na tabela *nome de pessoa*, que junto com o identificador *val_id* compõem a chave. Este valor, *val_id*, é incrementado a cada ocorrência de um novo nome. Para implementar uma propriedade dinâmica também são precisos campos com datas de *transação*, *validade* e *fim*. A *data de transação* é a data em que a propriedade mudou de valor. A *data de validade* é a data a partir da qual o novo valor irá valer e a *data de fim* é a data em que este valor passa a ser antigo, isto é, já foi inserida outra tupla com um novo valor para esta propriedade dinâmica. Quando é inserido uma tupla com o valor novo, este campo fica vazio.

Como exemplo de uma implementação de propriedades estáticas e dinâmicas, considere uma pessoa chamada Maria Nunes. Esta pessoa casa-se no dia 5 de março de 1995 e decide mudar o nome para Maria Nunes Vieira, incluindo o sobrenome do marido. No dia 15 de janeiro de 1997 é colocada a informação no banco de dados de uma futura troca de nome no dia 20 de fevereiro, quando esta pessoa retornará ao nome inicial devido a um divórcio.

Tabela 5: Nomes

oid	val_id	nome	transação	Fim
16	1	Maria Nunes	01/01/90	05/03/95
16	2	Maria Nunes Vieira	05/03/95	20/02/96
16	3	Maria Nunes	15/01/96	

Para procurar o nome atual desta pessoa identificada pelo valor 16, é preciso verificar o maior *val_id* para todas as ocorrências de *oid* igual a 16. Para extrair um histórico dos nomes que esta pessoa já possuiu, basta listar os nomes das tuplas que têm como *oid* o valor 16. Outra propriedade dinâmica presente em todas as entidades deste estudo de caso, é o *estado* em que se

encontram. Como o *nome de pessoa*, o *estado*, sendo uma propriedade dinâmica, também é representado separadamente.

7. Conclusões

Durante a existência de um sistema de banco de dados seu esquema conceitual pode mudar (evoluir) devido a modificações ocorridas na legislação, devido a modificações nos requisitos dos usuários e devido a mudanças nos requisitos dos dados. O objetivo principal deste artigo foi estudar a evolução de esquemas conceituais de bancos de dados, considerando bancos de dados temporais, quando envolvido o conceito de papel associado a um modelo de dados orientado a objetos.

No enfoque aqui apresentado a representação das diversas versões do esquema ficam armazenadas em um meta-banco de dados, também temporal. Bancos de dados temporais são utilizados para armazenar tanto a extensão do banco de dados como as diversas versões do esquema conceitual.

Inúmeras são as possíveis alterações em um esquema, gerando novas versões. Neste artigo foi mostrado que, se o modelo utilizado apresenta o conceito de papel, as alterações geralmente ocorrem neste nível, facilitando tanto a construção da nova versão do esquema quanto a necessidade de adaptação da extensão do banco de dados.

É visto, ainda, um exemplo de mapeamento de um modelo de dados orientado a objetos que apresenta o conceito de papel (TF-ORM) para um banco de dados comercial relacional, demonstrando desta forma que a utilização, na especificação do sistema de informação, de um modelo expressivamente poderoso como o TF-ORM não implica, necessariamente, na utilização de um SGBD que suporte especificamente este modelo de dados pois o mapeamento para um modelo de dados convencional é bastante simples. Possíveis alterações nos papéis do esquema utilizado e seus reflexos nos dados armazenados são exemplificados através deste exemplo.

Do ponto de vista de armazenamento de dados a grande disponibilidade de dispositivos de armazenamento de baixo custo viabilizam a necessidade de múltiplas tabelas no banco de dados. Quanto ao desempenho, a aplicação a ser atendida é de sistemas elaborados de consulta para apoio à decisão. Nenhum tipo de processamento de consultas, mesmo sobre grandes bases de dados, é muito demorado em um Pentium 200 MHz. O problema de desempenho encontra-se em aplicações com necessidade de processamento transacional, o que não é o caso da aplicação pretendida.

8. Referências Bibliográficas

- [Castilho 93] CASTILHO, J.M.V. de. A States-Space approach for database redesign. *Proceedings of the 12th International Conference on Entity-Relationship Approach*, December 1993, Dallas, Texas.
- [Clifford 95] CLIFFORD, J.; YUZHILIN, A. Recent Trends in Temporal Databases. *Proceedings of the International Workshop on Temporal Databases*, Zurich, Switzerland, 17-18 Sept, 1995. Great Britain: Springer-Verlag, 1995. 360p.
- [DeCastro 95] DE CASTRO, C.; GRANDI, F.; SCALAS, M.R. On Schema versioning, in temporal databases. *Recent Advances in Temporal Databases*, J. Clifford, A. Tuzhilian (eds), Springer Verlag, 1995. p.272-291.
- [Edelweiss 93a] EDELWEISS, N.; OLIVEIRA, J. P. M.; PERNICI, B. An Object-oriented temporal model. *Proceedings of the 5th International Conference on Advanced Information Systems Engineering - CAISE'93*, Paris, France, June 8-11, 1993. pp.397-415. (Lecture Notes in Computer Science n. 685).
- [Edelweiss 93b] EDELWEISS, N.; OLIVEIRA, J.P.M. CASTILHO, J.M.V. A Temporal Logic Language for Temporal Conditions Definition. *Proceedings of the 13th International Conference of the Chilean Computer Science Society*, La Serena, Chile, 14-16 outubro, 1993. pp.163-178.

- [Edelweiss 94] EDELWEISS, N.; OLIVEIRA, J.P.M.; PERNICI, B. An Object-oriented approach to a temporal query language. *Proceedings of the 5th Database and Expert Systems Applications Conference - DEXA'94*, Athens, Greece, Sept. 7-9, 1994. pp.225-235. (Lecture Notes in Computer Science 856).
- [Edelweiss 95] EDELWEISS, N.; OLIVEIRA, J. Palazzo.M.; CASTILHO, J.M.V. Evolução de Esquemas em Bancos de Dados Temporais. *Anais do XXII Simpósio Brasileiro de Software e Hardware (XXII SEMISH), XXI Conferencia Latinoamericana de Informatica (PANEL'95)*, Canela, RS, 31 jul - 4 ago, 1995. p.375-386.
- [Ewald 93] EWALD, Catherine A.; ORLOWSKA, Maria E. A Procedural approach to schema evolution. *Proceedings of the International Conference on Advanced Information Systems Engineering CAiSE'93*, 5., June 8-11, 1993, Paris, France, p.22-38. (Lecture Notes in Computer Science n.685).
- [Kim 95] KIM, W.S.; CHANG, C.C.; LIM, T.Y.; SHIN, Y.H. Temporal object-oriented data model for the schema modification. *Proceedings of the 4th International Conference on Database Systems for Advanced Applications*, Singapore, April 10-13, 1995.
- [Lerner 90] LERNER, B.S.; HABERMANN, A.N. Beyond schema evolution to database reorganization. *Proceedings of the International Conference on Object-Oriented Programming: Systems, Languages, and Applications - OOPSLA'90 and European Conference on Object-Oriented Programming - ECOOP'90*, Oct. 21-25, 1990, Ottawa, Canada, p.67-76.
- [McKenzie 90] MCKENZIE, E.; SNODGRASS, R. Schema evolution and the relational algebra. *Information Systems*, v.15, n.2, p.207-232, 1990.
- [Oliveira 95] OLIVEIRA, J. Palazzo M. et al. On the Implementation of an Object-Oriented Temporal Model using Object-Oriented and Relational DBMSs. *Workshop Proceedings of the 6th Database and Expert Systems Applications - DEXA'95*, London, United Kingdom, Sept. 1995. Norman Revell, A. Min Tjoa (Eds.). p.35-44.
- [Penney 87] PENNEY, D.J.; STEIN, J. Class modification in the GemStone Object-Oriented DBMS. *SIGPLAN Notices*, New York, v.22, n.12, p.111-7, Dec. 1987.
- [Pernici 90] PERNICI, B. Objects with Roles. *SIGOIS Bulletin*, v.11, n.2-3, p.205-15, 1990.
- [Roddick 92] RODDICK, J.F.; PATRICK, J.D. Temporal semantics in information systems - a survey. *Information Systems*, v.17, n.3, p.249-267.
- [Silva 94] SILVA, M.P.; LANZELLOTTE, R.S.G. Formalizando a evolução de esquemas em SGBDs orientados a objetos. *9º Simpósio Brasileiro de Banco de Dados*, S. Carlos, SP, 5-7 setembro, 1994. P.157-172.
- [Snodgrass 85] SNODGRASS, R; AHN, I. A Taxonomy of time in databases. *Proceedings of the ACM SIGMOD International Conference On Management Of Data*, May 28-31, 1985, Texas, p.236-46.
- [Tansel 93] TANSEL. A.U.; CLIFFORD, J.; GADIA, S.; JAJODIA, S.; SEGEV, A.; SNODGRASS, R. *Temporal Databases - Theory, Design, and Implementation*. Redwood City: Benjamin/Cummings, 1993. 633p.
- [Tsichritzis 87] TSICHRITZIS, D. et al. KNOs: Knowledge Acquisition, Dissemination, and Manipulation Objects. *ACM Transactions on Office Information Systems*, New York, v.5, n.1, p.96-112, Jan.1987.
- [Wieringa 91] WIERINGA, R.; De JONGE, W. *The identification of objects and roles - object identifiers revisited*. Technical Report IR-267, Vrije University, Amsterdam, Holland, Dec. 1991.
- [Zicari 91] ZICARI, R. A Framework for schema updates in an object-oriented database system. *7th Data Engineering*, Kobe, Japan, April 8-12, 1991. p.2-12.